Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1973

Natural modes and frequencies of free vibration of shells of revolution: analysis of a shell of linearly varying thickness - Satans-III: Theory and user's manual.

Nebiker, Ralph Robert.

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/16718

Downloaded from NPS Archive: Calhoun

http://www.nps.edu/library

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community. Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

NATURAL MODES AND FREQUENCIES OF
FREE VIBRATION OF SHELLS OF REVOLUTION

—

ANALYSIS OF A SHELL OF
LINEARLY VARYING THICKNESS

—

SATANS-III: THEORY AND USER'S MANUAL

by

Ralph Robert Nebiker

Thesis Advisor: R.E. Ball

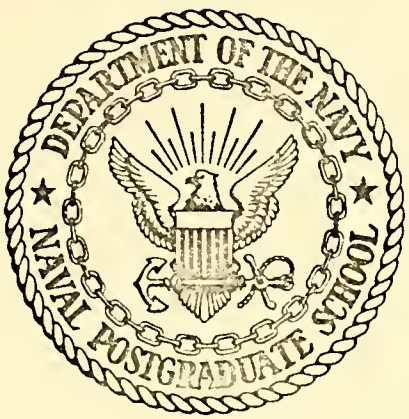
March 1973

Thesis
N3524

Approved for public release; distribution unlimited.

Library
Naval Postgraduate School
Monterey, Californ 93940

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

NATURAL MODES AND FREQUENCIES OF
FREE VIBRATION OF SHELLS OF REVOLUTION

—

ANALYSIS OF A SHELL OF
LINEARLY VARYING THICKNESS

—

SATANS-III: THEORY AND USER'S MANUAL

by

Ralph Robert Nebiker

Thesis Advisor: R.E. Ball

March 1973

Approved for public release; distribution unlimited.

Natural Modes and Frequencies of Free Vibration of Shells of Revolution

Analysis of a Shell of Linearly Varying Thickness

SATANS-III: Theory and User's Manual

by

Ralph Robert Nebiker
Lieutenant, United States Navy
B.S., Stevens Institute of Technology, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 1973

N3521
c.2

Library
Naval Postgraduate School
Monterey, California 93940

ABSTRACT

A digital computer program known as SATANS-III is developed as a modification to the program SATANS-I, incorporating the capability to perform free vibration analyses for the natural frequencies and modes of thin, elastic shells of revolution. The geometric and material properties of the shell may vary along the meridian. SATANS-III retains the SATANS-I capability of performing geometrically nonlinear static and dynamic analyses of arbitrarily loaded shells. The free vibration analysis employs the inverse iteration method with spectral shifts, including the capability to search a specific range of frequencies. Several example problems are solved to illustrate the program's validity and accuracy. A user's manual for SATANS-III is presented. In addition, the problem of handling a large, strongly banded, unsymmetric and occasionally singular stiffness matrix in conjunction with a singular mass matrix is treated. A discussion of several solution procedures with respect to this problem is given.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 12

II. BRIEF DESCRIPTIONS OF SATANS: VERSIONS I, II AND III ---------- 14

A. THE ORIGINAL SATANS ---------- 14

B. SATANS-I ---------- 15

C. SATANS-II ---------- 16

D. SATANS-III ---------- 16

III. SOLUTION PROCEDURE ---------- 18

A. SHELL COORDINATE SYSTEM ---------- 18

B. METHOD OF SOLUTION ---------- 18

C. STATIC ANALYSIS ---------- 28

D. DYNAMIC ANALYSIS ---------- 30

IV. DEVELOPMENT OF SATANS-III ---------- 31

A. FREE VIBRATION EQUATIONS ---------- 31

B. CHARACTERISTICS OF THE PROBLEM ---------- 32

1. Desired Results ---------- 32

2. Nature of the Problem ---------- 32

C. SOLUTION TO THE EIGENVALUE PROBLEM ---------- 34

D. TESTING OF SATANS-III ---------- 40

V. CONCLUSIONS ---------- 42

APPENDIX A OTHER SOLUTION METHODS ---------- 44

APPENDIX B NATURAL FREQUENCIES OF SOME SELECTED SHELLS ---------- 52

APPENDIX C CONVERTING SATANS-I TO SATANS-III
PART ONE: COMMON BLOCK MODIFICATIONS ---- 63

PART TWO: MODIFYING SATANS-I SUBROUTINES ---------- 71

PART THREE: NEW SUBROUTINES - THEORY AND FLOWCHARTS ---------- 76

APPENDIX D EXPERIENCE AND RECOMMENDATIONS ON USING SATANS-III ---------- 169

APPENDIX E SATANS-III INPUT CARD DATA GUIDE --------- 172

APPENDIX F INPUT AND OUTPUT
PART ONE: USER SUPPLIED SUBROUTINES ----- 182
PART TWO: SAMPLE PRINTOUT ---------- 184

APPENDIX G EXAMPLES OF USER SUPPLIED SUBROUTINES AND A LISTING OF SATANS-III ---------- 190

LIST OF REFERENCES ---------- 298

INITIAL DISTRIBUTION LIST ---------- 301

FORM DD 1473 ---------- 302

LIST OF TABLES

I. Minimum Natural Frequencies of a Thin Circular Cylindrical Shell ---------- 52

II. Natural Frequencies of a Thin Circular Cylindrical Shell ---------- 53

III. Minimum Natural Frequencies of a Freely Supported Circular Cylindrical Shell with and without Inplane Inertia ---------- 54

IV. Minimum Natural Frequencies of a Shell of Positive Gaussian Curvature ---------- 56

V. Effect of a Linearly Varying Thickness on Minimum Frequencies ---------- 58

VI. Shell Configurations ---------- 60

VII. Boundary Condition Specification ---------- 62

VIII. Contents of SATANS-III Common Blocks ---------- 64

IX. Some FORTRAN Variables ---------- 69

LIST OF FIGURES

1. Shell Geometry and Coordinate System ---------- 19
2. Positive Directions for Variables ---------- 20
3. System Stiffness Matrix, Load and Solution Vectors ---------- 26
4. Nonlinear Behavior Affecting Convergence -------- 29
5. Reduction of a Matrix to Hessenberg Form -------- 46
6. Minimum Natural Frequencies of a Freely Supported Circular Cylindrical Shell ---------- 55
7. Minimum Natural Frequencies of Three Cones ------ 57
8. Effect of a Linearly Varying Thickness on Minimum Natural Frequencies ---------- 59
9. Search Scheme Schematic ---------- 93
10. Subroutine VIBERS(*) Flowchart ---------- 97
11. Subroutine ITREAL(*) Flowchart ---------- 102
12. Subroutine ITRATE Flowchart ---------- 105
13. Subroutine ITCPLX(*) Flowchart ---------- 121
14. Subroutine RANGER(*) Flowchart ---------- 127
15. Subroutine ADJUST(IADJ,*,*) Flowchart ---------- 134
16. Subroutine BTTM(*) Flowchart ---------- 144
17. Subroutine TOPPER(*) Flowchart ---------- 151
18. Subroutine INITAA Flowchart ---------- 158
19. Subroutine START(K) Flowchart ---------- 163
20. Subroutine VECOUT(IND,*) Print-out Samples ------ 166

TABLE OF SYMBOLS

a	characteristic shell reference length
b	nondimensional inplane stiffness
d	nondimensional bending stiffness
E_y	Young's modulus
E_o	Reference Young's modulus
f	natural frequency, CPS or Hz
h	shell thickness
h_o	shell reference thickness
h_1	thickness at initial edge
h_2	thickness of final edge
ht	shell height at axis of revolution
M_s, M_θ, $M_{s\theta}$	bending and twisting moments per unit length
m	shell mass density, $lb\text{-}sec^2/in^4$
m_s, m_θ, $m_{s\theta}$	nondimensional bending and twisting moment series coefficients
N,n	Fourier series circumferential harmonic (wave number)
N_s, N_θ, $N_{s\theta}$	membrane forces per unit length
$\hat{N}_{s\theta}$	effective shear force
p	nondimensional pressure load series coefficient
$\hat{Q}_s$	effective transverse force
q, q_s, q_θ	normal, meridional and circumferential components of applied pressure loading
R_s, R_θ	principal radii of curvature
r	normal distance from the axis of shell

$\bar{r}$	normal distance from shell axis used in nondimensional frequency parameter Ξ
r_1	normal distance from shell axis at initial edge
r_2	normal distance from shell axis at final edge
s	meridional shell coordinate
S_T	total meridional shell length
T	dimensional time
T_o	reference time
t	nondimensional time
t_s, t_θ, $t_{s\theta}$	nondimensional membrane force series coefficients
t_T	nondimensional thermal membrane force series coefficients
U, V, W	displacements tangent to the meridian, tangent to the parallel circle and normal to the reference surface
u, v, w	nondimensional series coefficients of U, V, and W
α^r	r^{th} iteration Rayleigh quotient eigenvalue, $\frac{-1}{(\omega^r T_o)^2}$
β	spectral shift point
γ	ρ'/ρ
Δ	distance between finite difference stations = $\frac{S_T}{(a(KMAX - 1))}$
δt	nondimensional time interval
ζ	shell coordinate perpendicular to reference surface
Θ	shell circumferential coordinate

μ_i	nondimensional mass at i^{th} station $= \frac{a^2}{h_o^2 E_o^2 T_o^2} \int m \, d\zeta$
Ξ	nondimensional frequency parameter $= \omega\bar{r}\sqrt{\frac{m(1-\nu^2)}{E_y}}$
ν	Poisson's ratio
ξ	nondimensional meridional coordinate, s/a
ρ	nondimensional normal distance to shell axis, r/a
$\sum_{i=m,j}^{n}$	summation over i from m to n in increments of j
σ_o	reference stress level
$\Phi, \Phi_s, \Phi_\theta$	reference surface rotations
$\phi, \phi_s, \phi_\theta$	nondimensional series coefficients for reference surface rotations
ω	natural frequency, radians per second
ω_s, ω_θ	nondimensional curvatures, a/R_s, a/R_θ

MATRIX NOTATION

$\underline{A}, \underline{B}, \underline{\bar{B}}, \underline{C}, \underline{E}, \underline{F}, \underline{G}, \underline{\bar{G}}, \underline{H}, \underline{J}, \underline{P}$, DEE, DST	4x4 matrices. Elements of the matrices are defined in Refs. 1 and 2.
$\bar{e}, \bar{f}, g, \bar{g}$	1 x 4 load vectors. Elements are defined in Refs. 1 and 2.
K	system stiffness matrix
ℓ	1 x 4 boundary condition vector
M	system mass matrix
W^r	system solution vector before eigenvalue extraction

z	1 x 4 solution vector
Z^r	system solution vector after scaling and eigenvalue extraction.
$\overline{\Lambda},\overline{\Omega}$	4 x 4 boundary condition matrices
$\overline{\mu}$	4 x 4 mass matrix
$[\ ]$	m x n matrix
$\{\ \}$	column vector
$\llcorner\ \lrcorner$	row vector
$\lvert\ \rvert$	absolute value of a scalar
i=m(k)n	i is stepped from m to n in increments of k

SUPERSCRIPTS

r	r^{th} iteration
T	transpose of a matrix or vector
'	(prime) partial differentiation in the meridional direction
-1	inverse of a matrix

ACKNOWLEDGMENT

The author wishes to express his deepest gratitude to Dr. Robert E. Ball for his abiding faith, trust and understanding. Dr. Ball's insight and expert professional guidance were principle factors leading to the successful completion of this project.

I. INTRODUCTION

The free vibration analysis of shells of revolution is of major interest to the modern structural analyst since shells are used in the construction of everything from spacecraft to nuclear reactors. The modes of free vibration and the associated natural frequencies are of fundamental importance in understanding the dynamic behavior of shells.

This thesis is devoted to the incorporation of a free vibration analysis capability in the computer program known as SATANS (Static And Transient Analysis, Nonlinear, Shells). The development of SATANS is described in Refs. 1, 2 and 3. In Section II of this thesis a brief description and history of the three versions of SATANS (known as SATANS, SATANS-I and SATANS-II) is presented. In Section III a brief explanation of the solution procedure used in SATANS is given. In Section IV, SATANS-III is developed. SATANS-III is a modification of SATANS-I incorporating the capability of a free vibration analysis. Major consideration was given to: (1) matching the vibration analysis procedure to the present program without the necessity of reformulating the original program, (2) maintaining the capability of having the shell material and geometric properties vary along the meridian, (3) having assurance of completely covering a frequency range in a given

Fourier harmonic. The selection of the inverse iteration procedure with spectral shifts meets these goals. In addition, a backup capability to handle unusual situations, such as a pair of complex conjugate eigenvalues, is included.

To illustrate the validity of SATANS-III the results of six analyses of shells of uniform thickness are compared with published results in Appendix B. In addition new results showing the analysis of a shell of linearly varying thickness is also given in Appendix B.

In Appendix A there is a short discussion of the difficulties of the present problem of a large, strongly banded, unsymmetric and occasionally singular stiffness matrix in conjunction with a singular mass matrix using other solution procedures.

The coding details for converting SATANS-I to SATANS-III are given in Appendix C. Appendices D through G constitute the user's manual for SATANS-III.

II. BRIEF DESCRIPTIONS OF SATANS: VERSIONS I, II AND III

SATANS [Refs. 1,2] existed in three forms prior to this work; the original code and two later modifications, SATANS-I and SATANS-II [Ref. 3].

A. THE ORIGINAL SATANS

SATANS [Refs. 1,2] is a digital computer program capable of performing geometrically nonlinear static and dynamic analyses of arbitrarily loaded shells of revolution. It is based upon a combination of Fourier series for the circumferential behavior and finite differences for the meridional analysis. The conditions that the program is based upon are:

1. The shell material is isotropic and Poisson's ratio is constant.
2. The elastic modulus may vary through the shell thickness.
3. All material and geometric properties are axisymmetric and may vary along the shell meridian.
4. The boundaries may be free, elastically restrained, fixed or closed (pole).
5. The loading may be any combination of a temperature distribution, pressure and edge load.
6. The applied pressure and temperature distributions, and initial conditions in the case of a dynamic analysis, must be symmetric about a reference meridional plane.

7. The maximum number of circumferential Fourier harmonics, MAXM, used to describe the loading and solution, is 10. The maximum number of meridional finite difference stations, KMAX, is 200. The limit on the product of KMAX and MAXM is 200.

SATANS requires approximately 155 K-bytes of core storage on an IBM 360/67 using the FORTRAN-H compiler with optimization level 2.

B. SATANS-I

SATANS-I, developed by Ryan [Ref. 3], consists mainly of a rearrangement of the original program and the addition of an equipment independent plotting package capable of handling 100 points. The rearrangement allows approximately 99% of the code to be pre-compiled and linked internally and stored on rapid-access secondary storage (disk, cell, tape, drum, etc). This stored package is called by a simple six card main program. The plotting package enables the user to obtain plots of all input and output variables on the high speed line printer. In addition, solutions at 36, vice 6 for SATANS, meridians may be printed and/or plotted. The restrictions on KMAX and MAXM are the same as for SATANS.

SATANS-I requires approximately 206 K-bytes of storage on an IBM 360/67 using the FORTRAN-H compiler with optimization level 2. Execution time is increased over that of SATANS by .01 seconds for each 100 point plot generated.

C. SATANS-II

SATANS-II, also developed by Ryan [Ref. 3], is an extensive expansion of SATANS-I. The following improvements were incorporated:

1. The ability to handle arbitrary initial imperfections in the shape of the shell was introduced.

2. The restriction on the applied loads, that they be symmetric about a reference meridian, was removed.

3. The maximum number of circumferential Fourier harmonics describing the completely general loads and the solution was increased to 13, and the maximum number of meridional stations was reduced to 100. The restriction on the product of KMAX and MAXM was raised to 951.

SATANS-II is a large program requiring 426 K-bytes of main core on an IBM 360/67 using the FORTRAN-H compiler with optimization level 2.

D. SATANS-III

SATANS-III is a modification of SATANS-I which incorporates the ability to perform a free vibration analysis of shells of revolution. SATANS-III operates under the same conditions as SATANS and SATANS-I. The procedure used to determine the modes and frequencies is inverse iteration with spectral shifts. One of three modes of operation is specified by the user. The three modes are:

1. Determination of the lowest frequency in a user specified circumferential harmonic.

2. Determination of the frequency closest to a user specified frequency in a user specified circumferential harmonic (supplied shift point mode).

3. Frequency range search (circumferential harmonic and end points of the frequency range specified by the user).

The assumptions made in determining the natural modes and frequencies are:

1. No loads of any type.

2. Small deflections and rotations.

3. All nonlinear effects are ignored.

SATANS-III requires 234 K-bytes of main core on an IBM 360/67 using the FORTRAN-H compiler with optimization level 2.

III. SOLUTION PROCEDURE

The solution procedure used in all of the versions of SATANS is based on Sanders' field equations [Ref. 4], with translational inertia terms added, assuming small strains and moderately small, or small, rotations. The basic method of solution was developed by Budiansky and Radkowski [Ref. 5] for the linear static case. A brief presentation of the method of solution is presented here for continuity. For a more detailed explanation refer to Refs. 1 and 2.

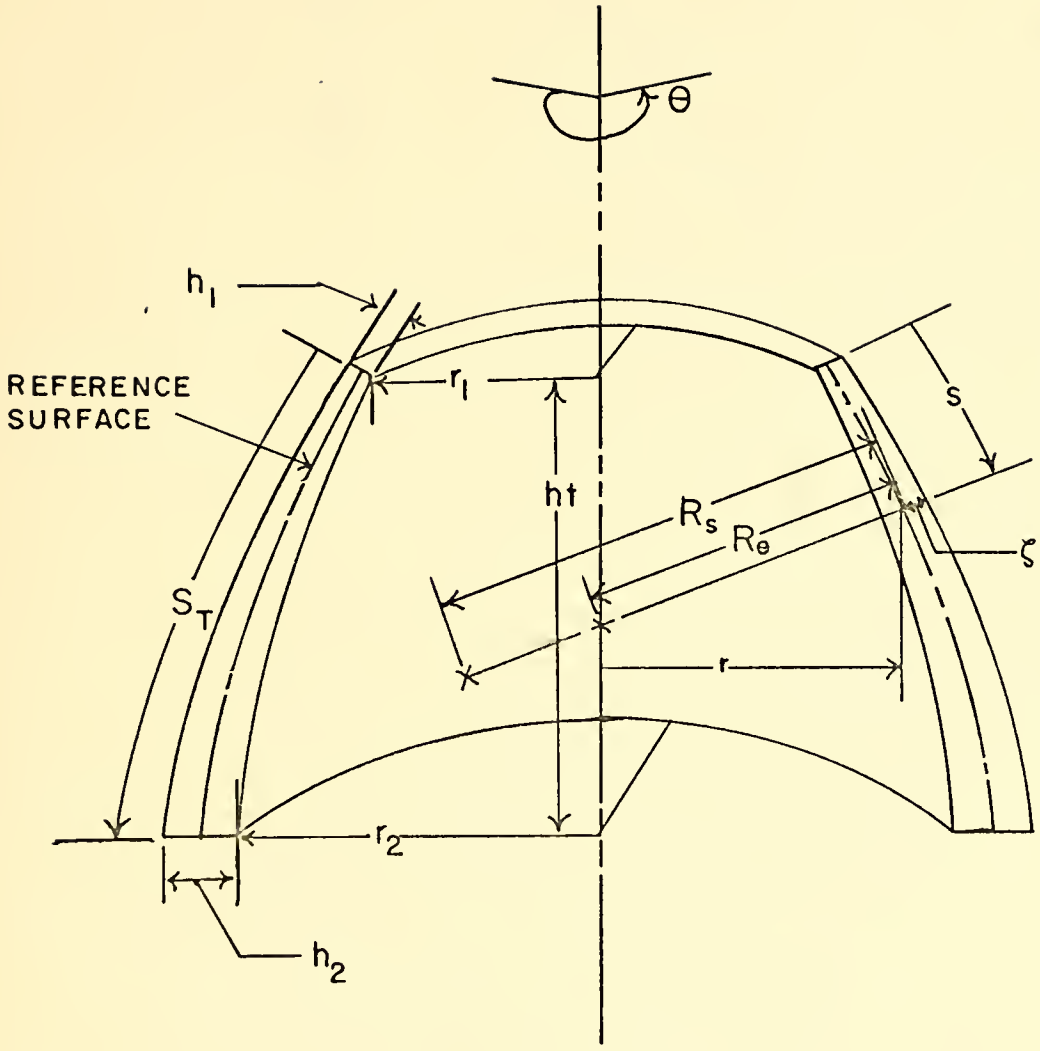
A. SHELL COORDINATE SYSTEM

Figure 1 depicts the orthogonal coordinate system used to describe the shell geometry. s is the distance along the meridian on the reference surface, θ is the circumferential angle, measured from some reference meridian, about the shell axis of revolution, and ζ is the coordinate through the thickness of the shell measured normal to the reference surface. The integral $\int E_y \zeta d\zeta = 0$ positions the reference surface, where E_y is Young's modulus.

Figure 2 depicts the positive directions for the displacements, rotations, forces, moments and loads.

B. METHOD OF SOLUTION

In the SATANS-I analysis the dependent variables in Sanders' nonlinear field equations are expanded into either a Fourier sine or cosine series in terms of $n\theta$, where n is



SHELL GEOMETRY AND COORDINATE SYSTEM

FIGURE 1

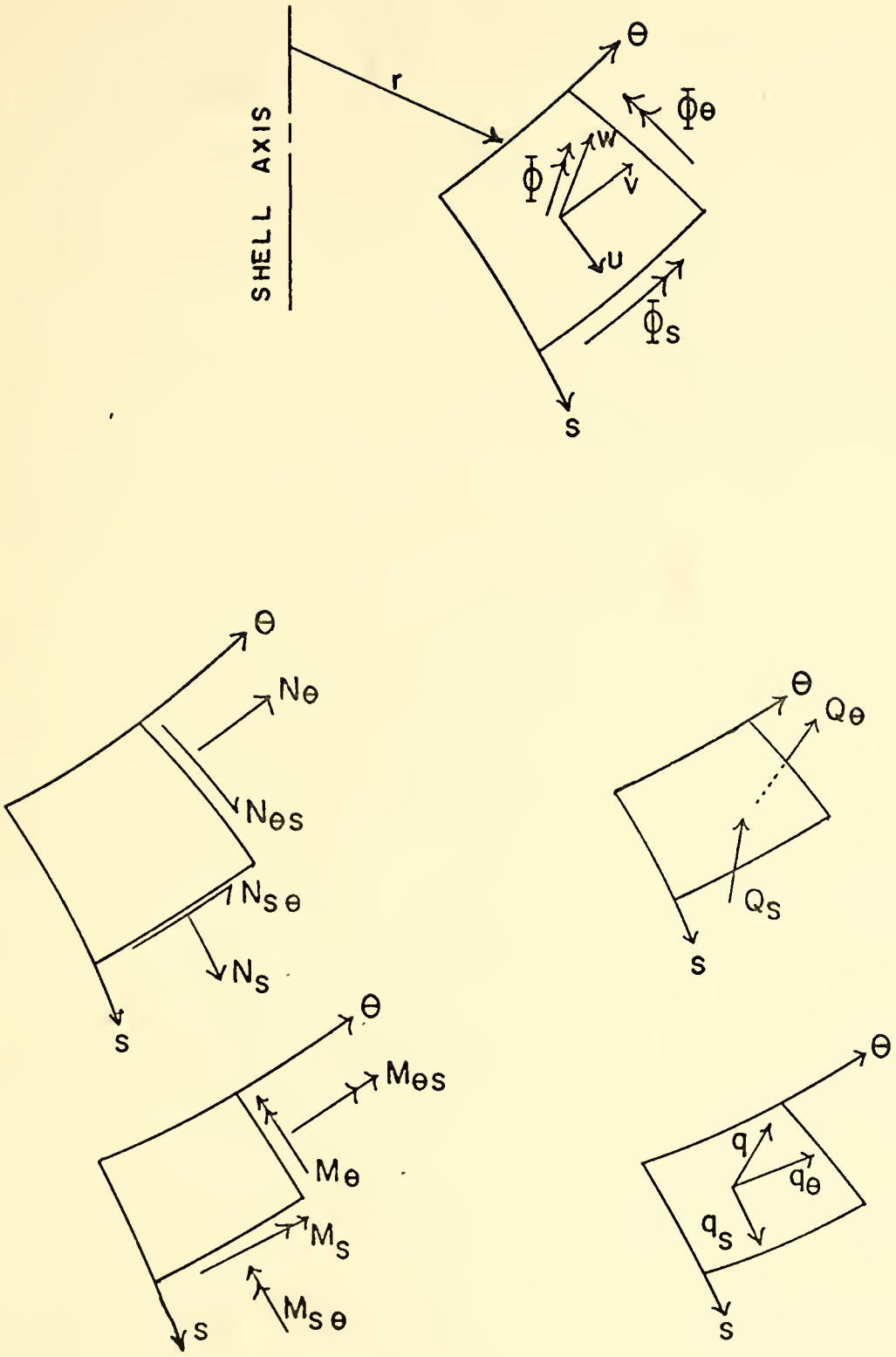


FIGURE 2 POSITIVE DIRECTIONS FOR VARIABLES

the Fourier harmonic index. This results in a complete set of governing equations for each Fourier harmonic n. The nonlinear terms are products of two series. These products are expressed as a single series, the coefficients of which are also series. In order to uncouple the sets of equations the "pseudo load" approach is used, i.e. the nonlinear coupling terms are treated as known quantities and are included with the loads on the right side of the equations. Thus the left-hand side of the equations is linear and the sets are uncoupled.

Following the method suggested in Ref. 5 each set of uncoupled equations (one set for each value of n) is expressed as four second order differential equations. The independent variables are the series coefficients $u^{(n)}$, $v^{(n)}$, $w^{(n)}$ and $m_s{}^{(n)}$, representing the nondimensional meridional, circumferential and normal displacements and the meridional bending moment respectively. For convenience the superscript n will be dropped henceforth.

For each value of n, the set of four second order differential equations may be written in matrix notation as

$$E\, z'' + F\, z' + G\, z = \bar{e} + \bar{\mu}\, \frac{\partial^2 z}{\partial t^2} \qquad (1)$$

where t is nondimensional time. Primes denote partial differentiation with respect to the nondimensional

meridional coordinate ξ. Vector $\bar{e}$ contains the load and nonlinear terms.

The solution vector z is

$$z = \begin{Bmatrix} u \\ v \\ w \\ m_s \end{Bmatrix}$$

The mass matrix $\bar{\mu}$ is

$$\bar{\mu} = \mu \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where the nondimensional mass μ is defined by the integral

$$\mu = \frac{a^2}{h_o E_o T_o^2} \int m \, d\zeta \qquad (2)$$

where m is the mass density of the shell material and a, h_o, E_o, and T_o are nondimensionalizing parameters. The elements of the 4 x 4 matrices E, F and G and the vector $\bar{e}$ are given in Ref. 2.

Equations (1) are solved numerically using spatial and timewise differencing schemes. In both the static and dynamic analysis the first and second derivatives of z

are approximated using the central finite difference schemes

$$z_i' = (z_{i+1} - z_{i-1})/2\Delta$$

$$z_i'' = (z_{i+1} - 2z_i + z_{i-1})/\Delta^2$$

where i denotes the station along the meridian and Δ is the uniform spacing between stations.

Houbolt's backward differencing scheme is used to approximate the second derivative with respect to time for the dynamic case.

$$\left.\frac{\partial^2 z}{\partial t^2}\right)_{i,j} = \frac{(2z_{i,j} - 5z_{i,j-1} + 4z_{i,j-2} - z_{i,j-3})}{(\delta t)^2}$$

The time step is denoted by j and δt is the nondimensional time interval.

The resulting set of four second order difference equations can be written in the matrix form

$$A_i\, z_{i+1,j} + \bar{B}_i\, z_{i,j} + C_i\, z_{i-1,j} = \bar{g}_{i,j} \qquad (3)$$

for i = 1,2,3,...,KMAX. The 4 x 4 matrices A, $\bar{B}$ and C and the vector $\bar{g}$ are given by

$$A_i = (2E_i/\Delta) + F_i$$

$$\bar{B}_i = (-4E_i/\Delta) + 2\Delta\bar{G}_i$$

$$C_i = (2E_i/\Delta) - F_i$$

$$\bar{G}_i = G_i - (2\bar{\mu}_i/(\delta t)^2)$$

$$\bar{g}_{i,j} = 2\Delta\bar{e}_{i,j} + \frac{2\Delta\bar{\mu}_i}{(\delta t)^2}(-5z_{i,j-1}+4z_{i,j-2}-z_{i,j-3})$$

In the static analysis j denotes the load step and $\mu_i = 0$.

The boundary conditions on z are entered on cards as the elements of two 4 x 4 matrices, $\bar{\Omega}$ and $\bar{\Lambda}$, and the column vector ℓ. The boundary conditions specify either the forces or the displacements at the shell boundary for all n considered.

The boundary condition matrix equation is

$$\bar{\Omega}\begin{Bmatrix} N_s \\ \hat{N}_{s\theta} \\ \hat{Q}_s \\ \Phi_s \end{Bmatrix} + \bar{\Lambda}\begin{Bmatrix} U \\ V \\ W \\ M_s \end{Bmatrix} = \ell$$

where all terms are identified in Figure 2. The ^ above $\hat{N}_s$ and $\hat{Q}_s$ denotes the effective edge variables. As an example, a clamped edge is specified by

$$\bar{\Omega} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \bar{\Lambda} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and

$$\ell = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

In order to convert the boundary conditions to finite difference form, the derivatives with respect to the meridional coordinate in the forces and rotations are replaced by the central finite difference schemes.

For each value of n the system of KMAX + 2 matrix equations may be written as

$$K\ Z = Y \qquad (4)$$

Figure 3 depicts the elements of the K matrix and the Z and Y vectors.* The first and last rows of K are the boundary condition equations. The elements of the 4 x 4 matrices H and J and the four element vector $\bar{f}$ are given

---

*There are KMAX stations over the length of the shell where Equation (3) applies. The first and last stations are at the initial and final edges. This leads to KMAX equilibrium equations. The boundary equations are also applied at the initial and final edges of the shell. Thus there are KMAX + 2 matrix equations. The application of Equation (1) and the boundary equations at the edges of the shell introduce one imaginary station off each edge of the shell. This results in KMAX + 2 unknown values of z. The vector Y consists of the KMAX values of $\bar{g}$ and the two terms at the initial and final positions representing the loads and the boundary condition values at the edges.

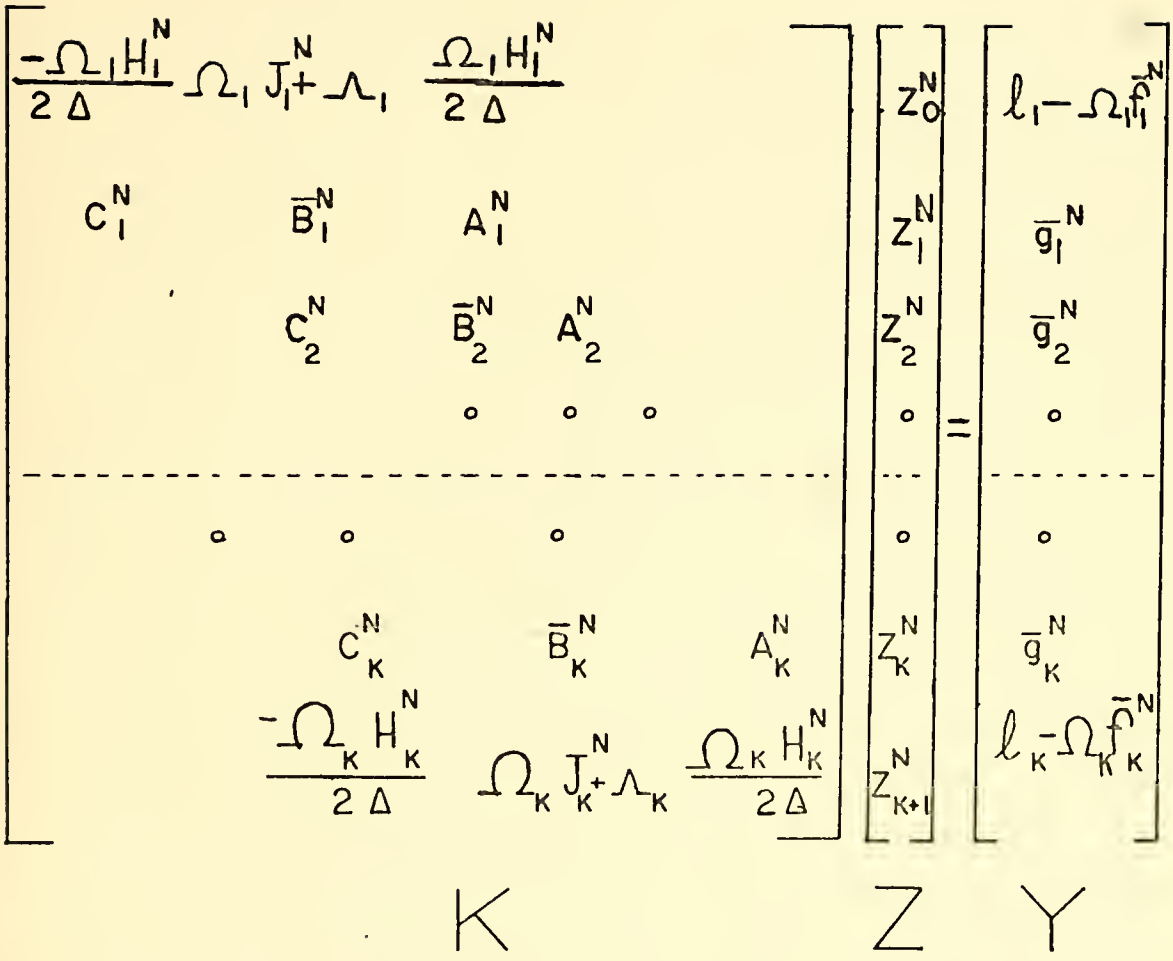


FIGURE 3 SYSTEM STIFFNESS MATRIX, LOAD AND SOLUTION VECTORS

in Ref. 2. The system is tridiagonal in the matrix sense and Potters' form of Gaussian elimination is used to solve for the $z_{i,j}$. For the forward pass

$$x_{i,j} = DEE_i \, \bar{g}_{i,j} - DST_i \, x_{i-1,j} \, .$$

For the backward pass

$$z_{i,j} = -P_i \, z_{i+1,j} + x_{i,j} \, .$$

The 4 x 4 matrices P_i, DEE_i and DST_i are independent of the load and solution and are determined only once. They are given by

$$DEE_i = (\bar{B}_i - C_i P_{i-1})^{-1}$$

$$P_i = DEE_i \, A_i$$

$$DST_i = DEE_i \, C_i$$

The initial value of $x_{i,j}$ on the forward pass and the initial value of $z_{k+1,j}$ on the backward pass are specially determined according to the boundary conditions. Special procedures are also invoked in the case of a shell with an initial and/or final pole [Ref. 2].

C. STATIC ANALYSIS

Figure 4 illustrates a typical load history and solution path for a static analysis. A solution to a monotonically increasing load is obtained at each load step in an iterative process. The load is incremented by DELOAD each step. A solution is obtained for each Fourier harmonic index considered. The pseudo loads are recomputed each iteration at each load step. The solution at each iteration is tested for convergence. There is no solution to estimate the pseudo loads on the first load step and the result after the first iteration is a linear solution. If convergence occurs, the load is incremented by the amount DELOAD, an estimate of the solution is made by a linear extrapolation of the last two solutions, and the iteration process is repeated.

If convergence is not obtained in a specified number of iterations, ITRMAX, the total load, ALOAD, is decremented by one DELOAD. DELOAD is reduced to 20% of its previous value and becomes the new incremental load. The iteration process is then repeated. The program terminates when DELOAD has been reduced to 20% of its previous value a specified number of times (LCHMAX), or when a specified number of load steps has been taken.

Because of the pseudo load approach, failure to converge to a solution may be due to either of two types of nonlinear behavior illustrated in Figure 4. Curve A experiences a

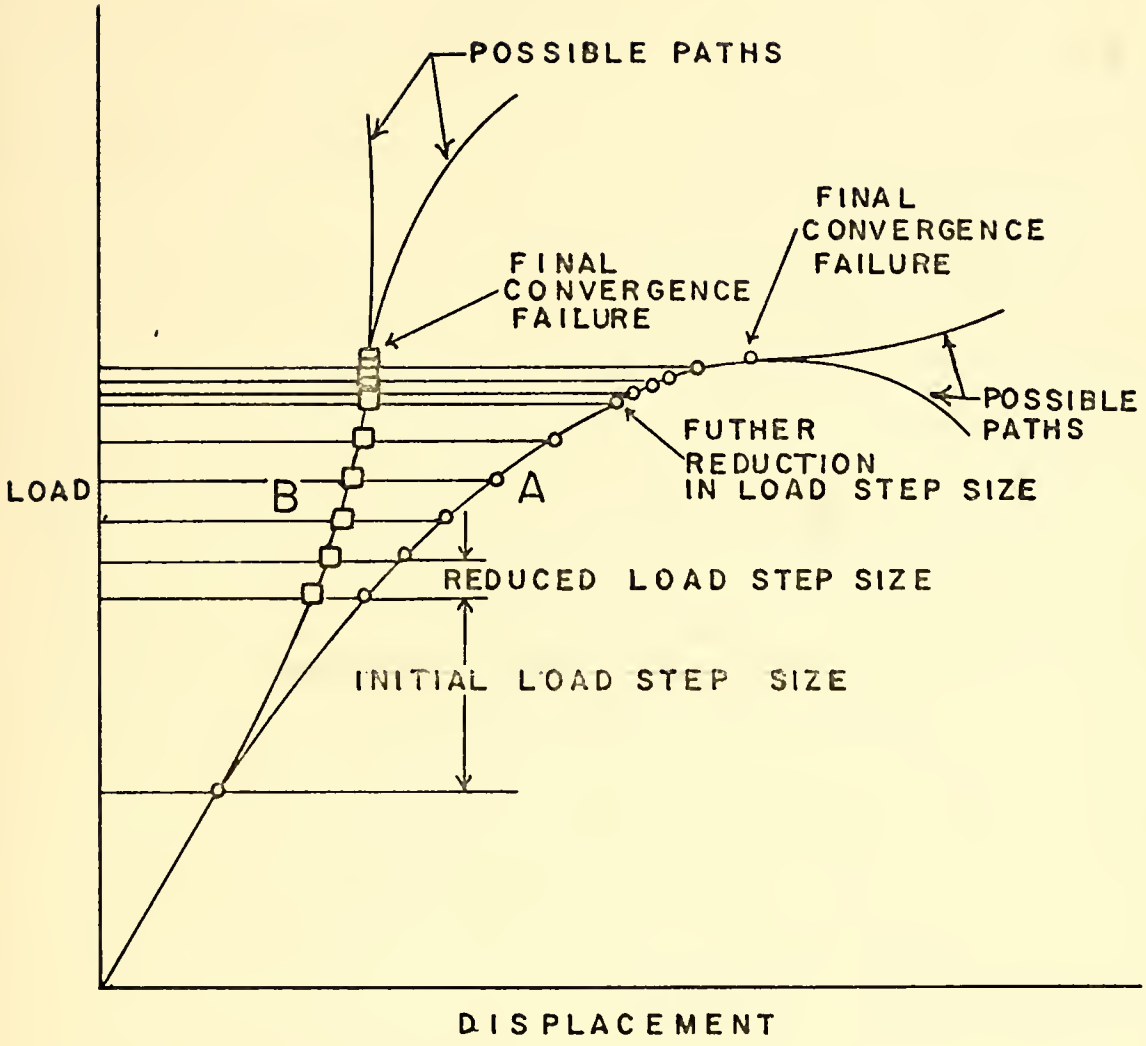


FIGURE 4 NONLINEAR BEHAVIOR AFFECTING CONVERENCE

maximum or inflection point, typical of a softening nonlinearity, causing convergence failure. Curve B experiences an ever increasing slope, typical of a stiffening nonlinearity, causing convergence failure.

D. DYNAMIC ANALYSIS

The applied load in the dynamic analysis is a function of the time step j. DELOAD is the incremental time δt. Initial conditions on z and $\partial z/\partial t$ are supplied in a user prepared subroutine, INITL. The procedure is essentially the same as the static analysis except the load step becomes the time step. The program terminates under one of two conditions. Either convergence is not obtained in the specified number of iterations, ITRMAX, or the maximum number of time increments, LSMAX, has been reached.

IV. DEVELOPMENT OF SATANS-III

A. FREE VIBRATION EQUATIONS

Recall the set of four second order differential equations written in Equation (1) was

$$E\,z'' + F\,z' + G\,z = \bar{e} + \bar{\mu}\,\frac{\partial^2 z}{\partial t^2} \qquad (1)$$

In the free vibration analysis the nonlinear terms are ignored and no loading exists. Hence the vector $\bar{e}$ does not exist under these conditions. Assuming harmonic motion, the solution vector $z(\xi,T)$ takes the form $e^{i\omega t T_o}\,z(\xi)$ where T_o is a time constant and ω is the natural frequency in radians per second. Therefore Equation (1) simplifies to

$$E\,z'' + F\,z' + G\,z = -\omega^2\,T_o{}^2\,\bar{\mu}\,z \qquad (5)$$

Using the same spatial finite differencing schemes used in SATANS-I, Equation (5) becomes

$$A_i\,z_{i+1} + \bar{B}_i\,z_i + C_i\,z_{i-1} = -\omega^2\,T_o{}^2\,\bar{\mu}_i\,2\Delta z_i \qquad (6)$$

for $i = 1,2,\ldots,KMAX$. The elements of the matrices A, $\bar{B}$, and C are the same as those for a static analysis. Equation (6) can be written in the matrix form of a standard

eigenvalue problem

$$K\ Z = -\omega^2\ T_o^2\ M\ Z \qquad (7)$$

where the K matrix is the same as the one shown in Figure 3. The mass matrix M will be described in the following section.

B. CHARACTERISTICS OF THE PROBLEM

There are three main areas to consider when incorporating a free vibration analysis capability in an existing program. Consideration must be given to the form of the desired results and the expected benefits of the vibration analysis. Consideration must also be given to both compatibility with the existing code and the nature of the problem itself.

1. Desired Results

Generally, only the lower frequencies in each Fourier harmonic are sought. The lowest 10 is a common figure. For example, Kalnins, in Reference 6, points out that there is a practical limit to which a linear theory correctly approximates the higher frequencies of a real shell. The mode shapes are also usually desired. In certain applications only a specific range of frequencies is of interest. Some assurance of finding all the frequencies in that range is desired.

2. Nature of the Problem

The mass matrix M is diagonal and singular and is of the form

$$M = 2\Delta \begin{bmatrix} 0 & & & & & & \\ & \bar{\mu}_1 & & & & & \\ & & \ddots & & & & \\ & & & \bar{\mu}_{i-1} & & & \\ & & & & \bar{\mu}_i & & \\ & & & & & \bar{\mu}_{i+1} & \\ & & & & & & \ddots \\ & & & & & & \bar{\mu}_{KMAX} \\ & & & & & & 0 \end{bmatrix}$$

When the meridional, circumferential and radial translational inertial forces are all considered

$$\bar{\mu}_i = \mu_i \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

If only the translational intertial forces in the radial direction are considered

$$\bar{\mu}_i = \mu_i \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The stiffness matrix, K, Figure 3, is unsymmetric and singular in some cases due to the boundary conditions. K has a bandwidth of 15, or 3 if considered in the matrix sense. When the number of meridional stations is the maximum of 200, K has an order of 808. If K were stored in the banded form taking no account of the zeroes in the band, 48 K-bytes of IBM 360/67 main core are required. Potters' elimination scheme requires only 38.4 K-bytes of main core. Becuase K is unsymmetric, there is no absolute guarantee, from a mathematical viewpoint, that the eigenvalues and eigenvectors are real. Some capability in this area is sought, both as a trouble shooting aid and as a complex root solver.

Ill-conditioning in the solution may result from several causes. Small separations between the eigenvalues, numerical conditioning of the K matrix and the loss of significance are some reasons that make eigenvalue/eigenvector determination difficult. Thus it is desirable to have a capability to handle such cases, but not to have to pay the price of determining every eigenvalue and eigenvector of the total system.

C. SOLUTION TO THE EIGENVALUE PROBLEM

The method of solution to the eigenvalue problem selected for inclusion in SATANS-I is inverse iteration with spectral shifts. This scheme melds with the existing code extremely well. In addition, a capability to handle special ill-

conditioning problems and complex roots has been included. A brief discussion of other possible solutions to the real unsymmetric eigenvalue problem appears in Appendix A.

The inverse iteration procedure is eminently suited for accurate determination of eigenvalues and eigenvectors [Refs. 7 and 8]. The amount of work involved is directly proportional to the number and accuracy of the eigenvalues and eigenvectors sought. The coding is simple and requires a very small increase in core. Lastly, the eigenvectors and eigenvalues are determined simultaneously. The basic scheme is well known [Refs. 7-14]. However, the scheme used in SATANS-III is slightly different and is explained below.

The basic iterative procedure is obtained by rewriting Equation (7) in the form

$$[K - \beta M]W^{r+1} = M\, Z^{r}$$

where

$$W^{r+1} = \alpha^{r+1}\, Z^{r+1}$$

and

$$\alpha = \frac{-1}{(\omega^2\, T_o^{\,2}) + \beta}$$

where r denotes the rth iteration and β is the spectral shift value.

a. The vector Z^r is assumed for the first iteration and afterward is computed using W^r. Prior to solving for W^{r+1} the mass matrix M must be premultiplied through the vector Z^r. This consists merely of multiplying the diagonal of $2\Delta\bar{\mu}_i$ through each $z_i{}^r$. Since the mass may vary along the meridian removal of this variable from the right hand side of the equation was desirable. This was accomplished by dividing each element of the A_i, $\bar{B}_i$ and C_i matrices by $2\Delta\mu_i$ at the time of their computation. These additional computations are only performed in the vibration analysis state of the program. The net result of this procedure is that the diagonal of $\bar{\mu}_i$ merely consists of ones and zeroes.

b. W^{r+1} is solved for by Potters' elimination scheme as in the original SATANS-I program.

c. α^{r+1} is determined from the "Rayleigh quotient type" computation

$$\alpha^{r+1} = \frac{(Z^r)^T W^{r+1}}{(Z^r)^T Z^r} \tag{8}$$

which may be thought of as resulting from

$$\frac{(Z^r)^T ([K - \beta M]^{-1} MZ^r)}{(Z^r)^T Z^r}$$

realizing, of course, that this is not the manner in which the computations are performed. In the limit, as r approaches infinity, Z^{r+1} approaches Z^r, when Z^r is real. This

computation also determines the sign of α^{r+1}. The vector Z^{r+1} is scaled such that the largest element before each iteration is unity. It sometimes happens in the first few iterations that α^{r+1} is smaller than the largest element of the vector, and scaling by the largest element prevents vector element build-up. In the limit, as r approaches infinity, α^{r+1} becomes equal to the largest element. Hence, if the convergence test is failed, steps a through c are repeated with

$$Z^{r+1} = \frac{W^{r+1}}{\max\ (W_i{}^{r+1})}$$

d. The primary convergence test is performed on α^{r+1}. The test is

$$\left| \frac{|\alpha^{r+1}| - |\alpha^{r}|}{\alpha^{r+1}} \right| = < \text{EPSVIB}.$$

The absolute value of α is used because due to the spectral shifts α may be negative. In that case α changes sign each iteration but converges in absolute value. A secondary convergence test is optional and is performed on the vector elements themselves. The vector test is more stringent than the primary convergence test and increases the computation time. The vector test consists of comparing the difference between two successive iterations with the maximum values of u, v, w and m_s, times EPSVEC, a convergence criterion, for the last iteration.

Recall that

$$Z = \left\{ \begin{array}{l} \cdot \\ \cdot \\ \cdot \\ z_{i-1} \\ z_i \\ z_{i+1} \\ \cdot \\ \cdot \\ \cdot \end{array} \right\} \quad \text{where i denotes the station,}$$

and

$$z_i = \left\{ \begin{array}{l} u_i \\ v_i \\ w_i \\ m_{s_i} \end{array} \right\} = Z_{i,k} \quad \text{where } k = 1(1)4$$

such that $Z_{i,1} = u$, $Z_{i,2} = v_i$ ... etc. for each station i. The vector test may then be stated as follows

$$Z_{i,k}^{r+1} - Z_{i,k}^{r} = < \max(Z_{i,k}^{r+1}) \times EPSVEC$$

for i = 1(1)KMAX2

k = 1(1)4.

EPSVEC = EPSVIB unless otherwise specified on input.

e. When convergence has occured one more iteration is performed and the vector is scaled using the value of α^{r+1} given by Equation (8). This yields an indication

of the quality of the solution by inspecting the largest element of the vector, which should be essentially 1.0.

f. The eigenvalue is obtained by taking the reciprocal of α^{r+1} and subtracting the spectral shift value, β.

$$(\omega^2 T_o^2)^{r+1} = \frac{-1}{(\alpha^{r+1})} - \beta$$

In the program the FORTRAN variables are

```
VREAL = (ONEDO/SCALEW) + ASHIFT
AREAL = SNGL(VREAL)
```

where AREAL and ASHIFT are single precision variables and VREAL, ONEDO and SCALEW are double precision variables.

g. If the program is unable to determine a real eigenvalue in the specified number of iterations (ITRMAX), control is passed to a backup routine, ITCPLX. This routine, which is described in Appendix C, Part Three, was developed to handle special ill-conditioning and non-convergence problems. The results of the routine are the two closest frequencies, real or complex, to the shift point. In addition, diagnostic messages are provided to aid in solving the problem at hand. These additional computations are only performed when the primary iteration routine fails to converge to a solution.

Some of the advantages of the selected scheme have been mentioned in the preceding paragraphs. Another is that iteration provides the ability to determine the frequencies

and mode shapes to almost any precision desired. The iterated vector is computed in double precision in order to handle cases of ill-conditioning and weakly determined frequencies, [Ref. 7]. This adds only slightly to the computation time and has paid off in the ease with which difficult frequencies are determined. Because the mode shapes are also determined, they require no additional calculations and yield an immediate answer to the question of which mode has actually been determined. The answer to this question is not always known in other methods [Ref. 15].

The iteration method avoids all manner of mathematical and numerical problems by actually performing the calculations of the solution procedure. The fact that the K matrix is unsymmetric and that both the K and M matrices are singular has no special impact on the solution procedure. There is no need to reformulate the problem or the program code, a major consideration in the present situation.

D. TESTING OF SATANS-III

The program has undergone considerable testing. Because of the size of SATANS-III, the inverse iteration scheme was tested in prototype form first. This included quite a bit of testing of the backup capability planned into the program in addition to the primary scheme [Refs. 7,8,16 and 17]. Appendix B tabulates six test cases for SATANS-III and compares them with published results. The agreement is excellent.

The test cases consist of determining the minimum frequencies of the lower circumferential harmonic (wave) numbers of three cones, a regular right circular cylindrical shell and a shell of positive gaussian curvature. There is also a determination of the lower spectrum of frequencies in each circumferential wave number for the right circular cylindrical shell. Lastly, there is an analysis of the effect of a linearly varying thickness on the minimum frequencies of the lower circumferential wave numbers of the right circular cylindrical shell.

V. CONCLUSIONS

Application of SATANS-III to many test cases has demonstrated that the inverse iteration technique with spectral shifts is a rapid, accurate and effective method for determining the natural modes and frequencies of the free vibration of shells of revolution. The simultaneous determination of the modes and frequencies provides an opportunity for determining if a mode has been missed, and the supplied shift point operation provides an easy method of determining the missed mode. The fact that sweeping of the vectors was not used in determining the higher frequency modes posed no special problem. In addition, not using vector sweeping allows the use of fewer mass points and more rapid calculation of some frequencies and modes. Dividing the variable mass into the component matrices of the K matrix proved to be an effective method of eliminating this variable from the right side of the vibration equations.

A study of the effect of a linearly varying thickness on the minimum frequencies of a circular cylindrical shell showed that as the percentage thickness of the lower edge over the upper edge increases the cylinder begins to exhibit behavior somewhat similar to that of a cone. The minimum frequency begins to appear in lower circumferential harmonics and the axial half wave number begins to rise.

The program experiences some difficulty with extremely thin shells, corners and very small radii of curvature. This might be alleviated by using a finer grid (more grid points) or as a further modification the incorporation of a variable grid capability.

Another further modification needed is the addition of preloads to the shell. Many of the methods used to determine the modes and frequencies of preloaded shells require accurate initial estimates of the solution to start the procedure. SATANS-III provides these estimates. It is felt that a preload capability can be incorporated into SATANS-III and still keep the program an in-core routine requiring only two core boxes.

APPENDIX A

OTHER SOLUTION METHODS

There are primarily two approaches to the unsymmetric matrix eigenvalue problem. The transformation methods are characterized by the operations, finite or infinite in number, being performed on the matrix.* The iterative methods are characterized by the operations being performed on the solution vector rather than the matrix.

There are basically four transformation methods capable of handling a real, unsymmetric matrix eigenvalue problem. Lanzcos' method, the QR and the LR algorithms, are well known [Refs. 7-12]. The QZ algorithm recently developed at Stanford University [Ref. 18] is based on the QR algorithm and can handle a problem in the form $Ax = \lambda Bx$ with A and B unsymmetric and complex. They all suffer, with respect to the present problem, from the same deficiencies. For example, the greatest portion of the computation time is spent before the first eigenvalue is extracted and they require that the order and size of the matrix be the same. For the case of an unsymmetric banded matrix enough core storage must be set aside for at least half the matrix.

---

*The general problem $Ax = \lambda Bx$ may also be thought of as $Kz = \lambda z$. In the present problem, the full matrix form of K would require 2,560 K-bytes of core storage on an IBM 360/37.

Lanzcos' method for the unsymmetric case requires core storage for the generation of two sets of biorthogonal vectors in addition to the matrix itself. It is believed that this method would require the greatest storage and computation time. The LR algorithm without interchanges maintains the banded form but suffers from excessive computation time for closely spaced eigenvalues. In addition, the LR algorithm may be numerically unstable in some cases.

By far the most popular and successful transformation method in general use is the QR (and now also the QZ) algorithm. It is stable, fast, accurate and can handle complex numbers. The IMSL group [Ref. 19] is dedicated to speed and accuracy, and virtually all of their routines are variations of the QR and QZ algorithms. The QR algorithm without reduction to Hessenberg form requires a comparatively large number of calculations. Consequently the matrices are generally reduced to Hessenberg form which considerably reduces the computation time required to reach convergence. Again the main objections with respect to the problem considered here is computation time or core storage. Wilkinson [Ref. 7] states "If the LR algorithm with interchanges (which is stable) or the QR algorithm is used the band form above the main diagonal is gradually destroyed." This is also true of any method that employs a transformation to Hessenberg form. See Figure 5.

Another disadvantage of the transformation methods is the necessity of determining the eigenvectors in a separate operation from the natural frequencies.

REDUCTION OF A MATRIX TO HESSENBERG FORM

GIVENS' METHOD

INPUT MATRIX

3.000	2.000	1.000	0.0	0.0	0.0	0.0	0.0
4.000	3.000	2.000	1.000	0.0	0.0	0.0	0.0
5.000	4.000	3.000	2.000	1.000	0.0	0.0	0.0
0.0	5.000	4.000	3.000	2.000	1.000	0.0	0.0
0.0	0.0	5.000	4.000	3.000	2.000	1.000	0.0
0.0	0.0	0.0	5.000	4.000	3.000	2.000	1.000
0.0	0.0	0.0	0.0	5.000	4.000	3.000	2.000
0.0	0.0	0.0	0.0	0.0	5.000	4.000	3.000

MAJOR STEP NUMBER 1

3.000	2.030	-0.937	0.0	0.0	0.0	0.0	0.0
6.403	5.927	-1.659	2.186	0.781	0.0	0.0	0.0
0.0	0.341	0.073	0.469	0.625	0.0	0.0	0.0
0.0	6.247	-1.406	3.000	2.000	1.000	0.0	0.0
0.0	3.904	3.123	4.000	3.000	2.000	1.000	0.0
0.0	0.0	0.0	5.000	4.000	3.000	2.000	1.000
0.0	0.0	0.0	0.0	5.000	4.000	3.000	2.000
0.0	0.0	0.0	0.0	0.0	5.000	4.000	3.000

MAJOR STEP NUMBER 2

3.000	2.030	-0.043	0.936	0.027	0.0	0.0	0.0
6.403	5.927	2.189	1.775	-0.445	0.0	0.0	0.0
0.0	7.375	5.740	-0.210	0.321	1.906	0.529	0.0
0.0	0.0	-0.537	0.133	-0.271	0.055	0.0	0.0
0.0	0.0	2.455	-3.288	0.201	1.168	0.848	0.0
0.0	0.0	6.353	0.273	0.750	3.000	2.000	1.000
0.0	0.0	2.647	0.0	4.242	4.000	3.000	2.000
0.0	0.0	0.0	0.0	0.0	5.000	4.000	3.000

FIGURE 5

MAJOR STEP NUMBER 3

3.000	2.030	-0.043	-0.059	-0.920	0.161	0.023	0.0
6.403	5.927	2.189	-0.279	-1.639	0.757	0.108	0.0
0.0	7.375	5.740	1.967	0.137	0.367	-0.194	0.0
0.0	0.0	7.327	5.790	0.369	-0.605	1.084	1.590
0.0	0.0	0.0	-0.296	-0.607	-0.200	-0.080	0.0
0.0	0.0	0.0	-0.181	-3.050	0.602	0.033	0.368
0.0	0.0	0.0	4.163	-1.115	-2.320	0.548	1.529
0.0	0.0	0.0	5.781	0.0	1.839	2.050	3.000

MAJOR STEP NUMBER 4

3.000	2.030	-0.043	-0.059	0.047	-0.619	-0.695	-0.066
6.403	5.927	2.189	-0.279	0.112	-1.503	-0.988	-0.155
0.0	7.375	5.740	1.967	-0.128	-0.242	0.291	0.178
0.0	0.0	7.327	5.790	1.921	0.709	0.088	0.056
0.0	0.0	0.0	7.132	3.864	-0.397	-0.233	0.312
0.0	0.0	0.0	0.0	-0.374	1.720	1.611	-0.018
0.0	0.0	0.0	0.0	0.348	-1.108	-1.893	0.063
0.0	0.0	0.0	0.0	0.767	-2.127	2.250	-0.148

MAJOR STEP NUMBER 5

3.000	2.030	-0.043	-0.059	0.047	-0.066	0.930	-0.019
6.403	5.927	2.189	-0.279	0.112	0.108	1.747	-0.441
0.0	7.375	5.740	1.967	-0.128	0.356	-0.049	-0.214
0.0	0.0	7.327	5.790	1.921	-0.208	-0.547	0.413
0.0	0.0	0.0	7.132	3.864	0.333	0.441	0.063
0.0	0.0	0.0	0.0	0.921	1.285	1.595	-2.098
0.0	0.0	0.0	0.0	0.0	0.227	0.034	-0.402
0.0	0.0	0.0	0.0	0.0	0.936	-2.757	-1.639

MAJOR STEP NUMBER 6

3.000	2.030	-0.043	-0.059	0.047	-0.066	0.200	-0.909
6.403	5.927	2.189	-0.279	0.112	0.108	-0.018	-1.802
0.0	7.375	5.740	1.967	-0.128	0.356	-0.220	-0.003
0.0	0.0	7.327	5.790	1.921	-0.208	0.273	0.629
0.0	0.0	0.0	7.132	3.864	0.333	0.165	-0.414
0.0	0.0	0.0	0.0	0.921	1.285	-1.663	-2.044
0.0	0.0	0.0	0.0	0.0	0.963	-2.270	2.199
0.0	0.0	0.0	0.0	0.0	0.0	-0.156	0.664

FIGURE 5 CONTINUED

```
C*************************************************************************
C                                                                        *
C      REDUCTION OF A MARIX TO HESSENBERG FORM BY GIVENS' METHOD         *
C                                                                        *
C*************************************************************************
      IMPLICIT INTEGER(R)
      DIMENSION A(8,8)
      READ(5,8) A
      WRITE(6,11)
      WRITE(6,9) ((A(II,JJ),JJ=1,8),II=1,8)
      DO 7 R = 1,6
      RP1 =  R + 1
      RP2 = R + 2
      DO  6 I = RP2,8
      IF(ABS(A(RP1,R))+ABS(A(I,R))) 2,1,2
    1 X = 0.0
C     CSX = 1.0
C     SNX = 0.0
      A(RP1,R) = X
      A(I,R) = 0.0
      GO TO  6
    2 X = SQRT(A(RP1,R)*A(RP1,R) + A(I,R)*A(I,R))
      CSX = A(RP1,R)/X
      SNX = A(I,R)/X
      A(RP1,R) = X
      A(I,R) = 0.0
      DO 4 J = RP1,8
      TEMP = CSX*A(RP1,J) + SNX*A(I,J)
      A(I,J) = CSX*A(I,J) - SNX*A(RP1,J)
      A(RP1,J) = TEMP
    4 CONTINUE
      DO 5 J = 1,8
      TEMP = CSX*A(J,RP1) + SNX*A(J,I)
      A(J,I) = CSX*A(J,I) - SNX*A(J,RP1)
      A(J,RP1) = TEMP
    5 CONTINUE
    6 CONTINUE
      WRITE(6,10) R
      WRITE(6,9) ((A(II,JJ),JJ=1,8),II=1,8)
    7 CONTINUE
    8 FORMAT(64F1.0)
    9 FORMAT(' ',30X,8F9.3)
   10 FORMAT('0',30X,'MAJOR STEP NUMBER',I2/)
   11 FORMAT(' ',46X,'REDUCTION OF A MATRIX TO HESSENBERG FORM'//60X,'GI
     1VENS'' METHOD'//31X,'INPUT MATRIX'//)
      STOP
      END
```

FIGURE 5 CONTINUED

An approach taken by some [Ref. 20] is to cast the problem into a form that is easily handled by one of the more popular methods. The difficulty here lies in that many of the operations depend on at least one of the matrices in the general problem $Ax = \lambda Bx$ being nonsingular in all circumstances. This was not the case with the present problem. It was felt that the amount of work involved in either reformulating the problem or reducing the matrices to a form that could be transformed easily involved too much computational effort.

A method classed as iterative, but one that does not operate on the vector, is that of evaluating a determinant. This has been applied to the present problem by other researchers [Refs. 6, 21, 22]. One of the difficulties with this method is that one or two initial guesses at the eigenvalue must be made to start the process. The better the guess the faster the convergence. Another is that, unlike the inverse iteration process, there is no inherent determination of the eigenvalue spectrum under consideration. Having no a priori knowledge of the solution to a particular problem it is possible to miss a mode and not realize it [Ref. 15]. This method also requires separate determination of the eigenvectors.

When the inverse iteration procedure was chosen as the solution procedure it was also decided that orthogonality and sweeping of the vectors would not be employed. Because the K matrix is never determined explicitly, the use of a

sweeping matrix was not considered. Sweeping, if it would be employed, would have to operate solely on the vectors [Ref. 23]. Because the K matrix is unsymmetric the right hand vectors (mode shapes) are not orthogonal through K or M, although they are independent. The vectors that are orthogonal are the right and left hand vectors. If modal sweeping is used a total of 6 K-bytes of main core is required for each vector swept (3 K-bytes per vector; two vectors per mode). An increase in computation time also results from the sweeping operation.

Orthogonality relationships for the mode shapes can be developed in the continuous theory, but if an unsymmetric K matrix results from the implementation of the theory then the right hand vectors are not, in fact, orthogonal. In some situations the right hand vectors have been treated as if they were orthogonal when they are not [Refs. 13, 24]. The critical factor, in this case, is the number of mass points chosen to approximate the continuous system. As the number of mass points increases the approximation to the correct mode shapes (which are orthogonal) becomes more accurate. For this condition the sweeping operation becomes effective, but the computation time also increases. With a few mass points the eigenvalues may be accurately determined and the computation time reduced considerably. However, the sweeping operation then serves only to introduce unwanted errors. It was felt that to effectively employ sweeping would result in a large computation time and core requirement

that could not be justified by the results. In addition, in some of the cases investigated using sweeping of the right hand vectors as if they were orthogonal, there seemed to be no immediately apparent correlation between an increase in the number of mass points and the resultant increase in the quality of the pseudo orthogonality of the right hand vectors.

APPENDIX B

NATURAL FREQUENCIES OF SOME SELECTED SHELLS

TABLE I

MINIMUM NATURAL FREQUENCIES (Hz) OF A THIN CIRCULAR CYLINDRICAL SHELL(1)

Circumferential Wave Number, n	Ref. 25	Ref. 26	SATANS-III
1		1446.0	1446.2
2	633.5	634.5	635.5
3	326.7	327.5	328.2
4	202.3	203.8	203.7
5	159.9	162.6	160.9
6	168.0	171.6	168.7
7	206.0	210.1	206.5
8	261.0	265.4	261.6
9		331.9	328.0
10	403.4	408.0	404.1
11		492.8	488.8
12	581.1	585.5	582.0
13		687.3	683.4
14	791.8	796.9	793.0
15		914.7	911.0
16		1041.0	1036.9
17		1175.0	1171.1
18		1317.0	1313.5
19		1467.0	1463.9
20		1626.0	1622.5

(1) Configuration 1, Table VI.

TABLE II

NATURAL FREQUENCIES (Hz) OF A THIN CIRCULAR CYLINDRICAL SHELL(1)

Circum. Wave Number, n	Axial Half-Wave Number, m											
	m = 1			m = 2			m = 3			m = 4		
	A	B	C	A	B	C	A	B	C	A	B	C
4	202.3	203.8	203.7	696.1	697.5	698.0						
5	159.9	162.6	160.9	483.0	484.6	484.6	960.6	962.5	963.3			
6	168.0	171.6	168.7	370.5	372.7	371.9	724.0	725.9	726.5			
7	206.0	210.1	206.5	325.1	328.1	326.3	580.8	583.2	583.2			
8	261.0	265.4	261.6	329.2	332.9	330.2	506.1	509.0	508.3	768.3	771.0	771.2
10	403.4	408.0	404.1	429.2	433.7	430.1	506.6	510.7	508.4	649.9	653.6	653.1
12	581.1	585.9	582.0	594.9	599.7	596.0	632.3	637.1	633.9	706.4	711.1	709.0
14	791.8	796.9	793.0	801.8	806.9	803.0	824.7	829.9	826.2	867.6	872.7	869.7

(1) Configuration 1, Table VI.
(A) Reference 25
(B) Reference 26
(C) SATANS-III

TABLE III

MINIMUM NATURAL FREQUENCIES (Hz) OF A FREELY SUPPORTED CIRCULAR CYLINDRICAL SHELL(1)

Circum. Wave Number, n	With Inplane Inertia DIAG = (1,1,1,0) f	Ξ(2)	Without Inplane Inertia DIAG = (0,0,1,0) f	Ξ
1	125.613	0.372	168.177	0.498
2	61.26	.182	69.24	.285
3	33.11	.0981	35.09	.104
4	19.92	.0590	20.58	.0610
5	13.62	.0403	13.90	.0412
6	10.07	.0298	10.21	.0303
7	8.494	.0252	8.583	.0254
8	8.285	.0245	8.351	.0247
9	9.057	.0268	9.114	.0270
10	10.33	.0306	10.38	.0308
11	12.08	.0358	12.13	.0359
12	14.24	.0422	14.29	.0423
13	16.58	.0491	16.63	.0493
14	19.16	.0568	19.21	.0569
15	21.96	.0651	22.01	.0652
16	24.94	.0739	24.99	.0740
18	31.56	.0935	31.61	.0936
20	38.95	.115	39.00	.115
22	47.14	.140	47.19	.140
24	56.12	.166	56.10	.166
26	65.87	.195	65.91	.195
28	76.40	.226	76.41	.226

(1) Configuration 2, Table VI.

(2) Dimensionless Frequency Parameter $\Xi = \omega \bar{r} \sqrt{\frac{m(1-\nu^2)}{E_y}}$

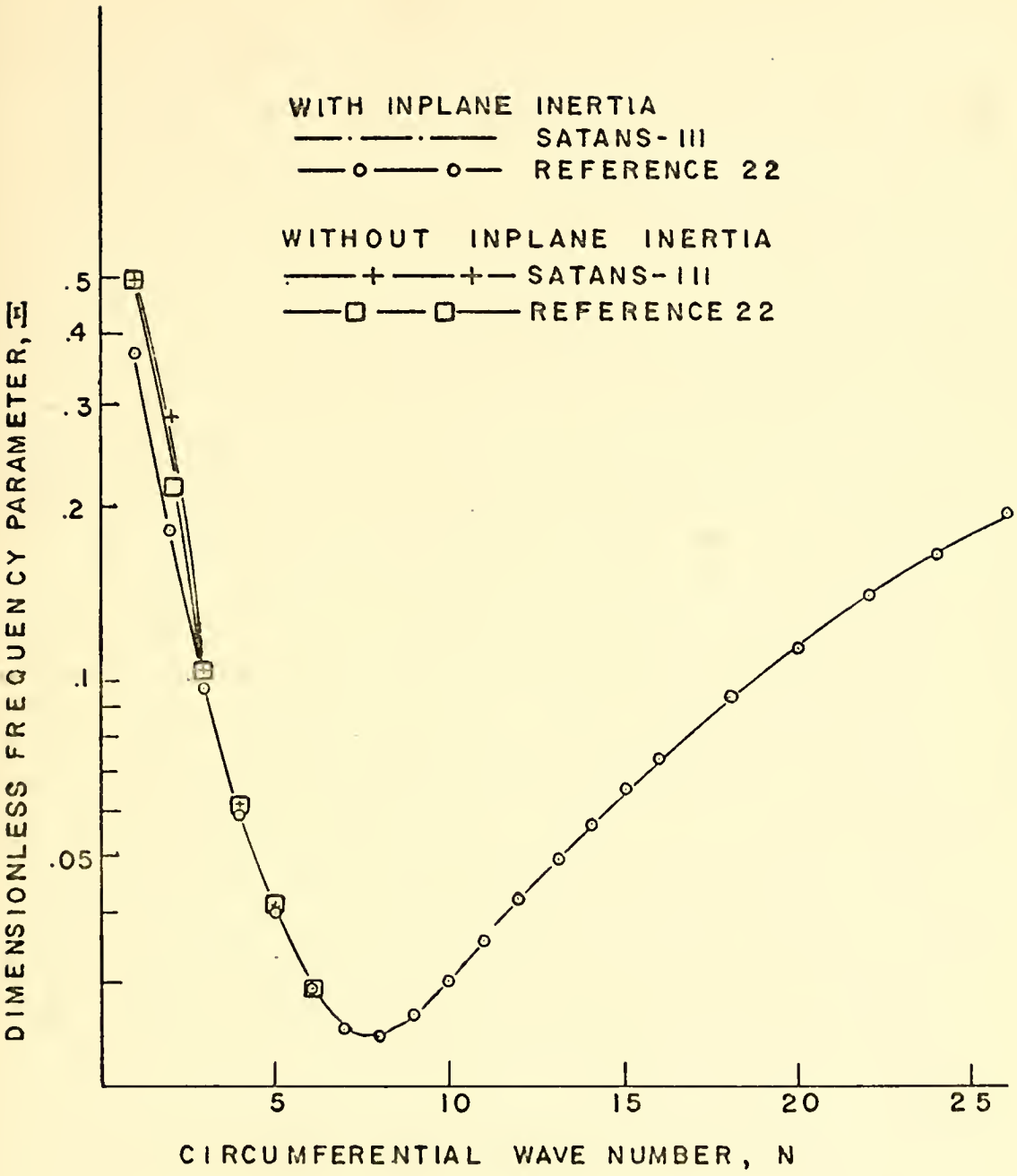


FIGURE 6 MINIMUM NATURAL FREQUENCIES OF A FREELY SUPPORTED CIRCULAR CYLINDRICAL SHELL CONFIGURATION 2, TABLE VI

TABLE IV

MINIMUM NATURAL FREQUENCIES OF A SHELL OF POSITIVE GAUSSIAN CURVATURE(1)

Dimensionless Frequency Parameter

$$\Xi = \omega \bar{r} \sqrt{\frac{m(1-\nu^2)}{E_y}}$$

Circumferential Wave Number, n	Ref. 27	Ref. 22	SATANS-III
1	0.411	0.412	0.411
2	.360	.362	.360
3	.340	.340	.340
4	.331	.331	.331
5	.327	.327	.327
6	.324	.324	.324
7	.323	.322	.323
8	.322	.321	.322
9	.321	.321	.321
10	.321	.321	.321

(1) Configuration 3, Table VI.

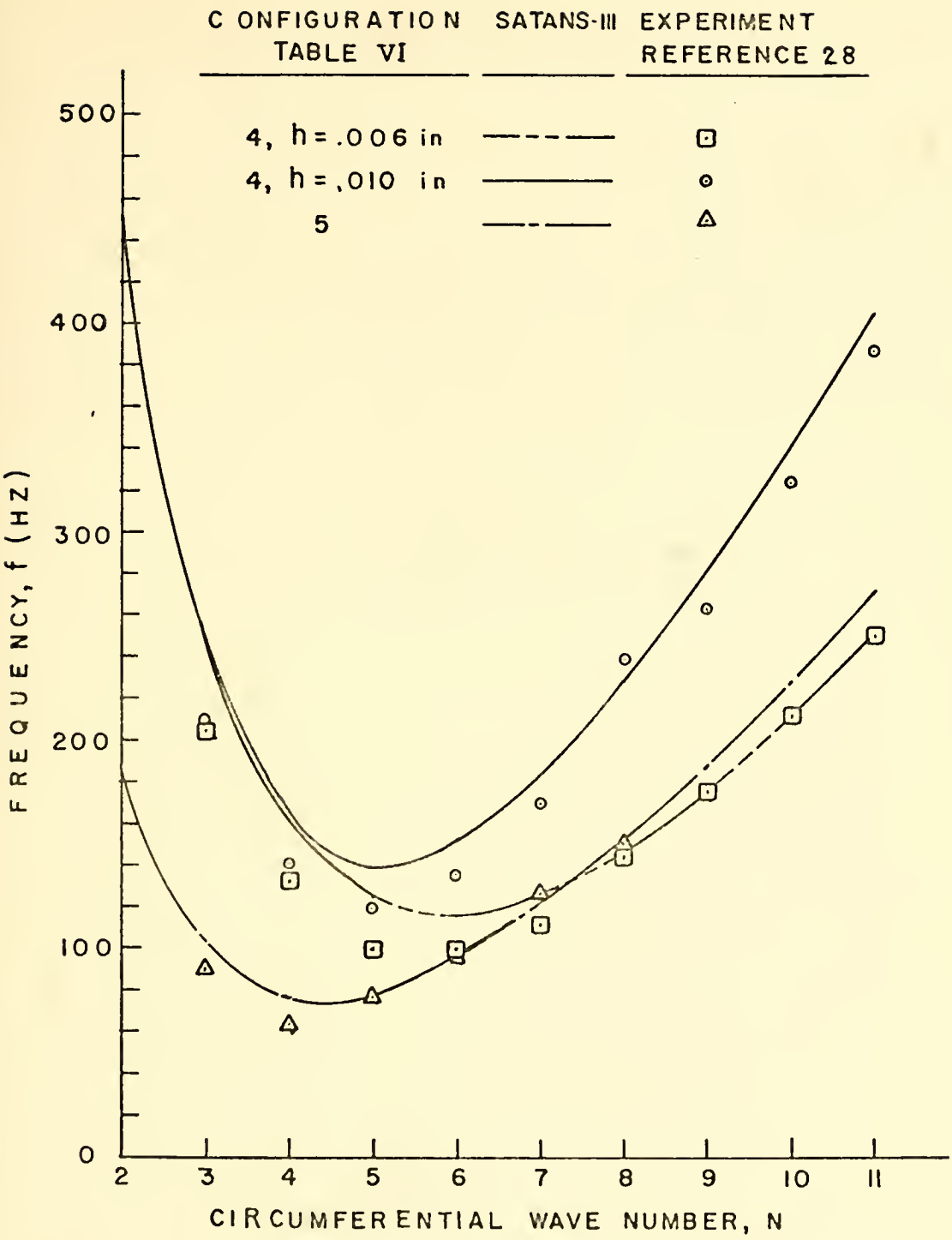


FIGURE 7

MINIMUM NATURAL FREQUENCIES OF 3 CONES

TABLE V

EFFECT OF A LINEARLY VARYING THICKNESS ON MINIMUM FREQUENCIES (Hz)(1)

$$h = .1(x - 1.)(s/S_T) + .1$$

Circum. Wave Number, n	x = 1		x = 2		x = 4		x = 7		x = 10	
	f	m(2)	f	m	f	m	f	m	f	m
1	125.6	1	122.1	1	112.7	1	104.4	1	99.01	1
2	61.26	1	60.72	1	58.44	1	56.09	1	54.28	1
3	33.11	1	32.94	1	32.12	1	31.30	1	30.71	1
4	19.92	1	20.19	1	20.02	1	20.26	1	20.81	1
5	13.62	1	13.85	1	14.80	1	16.43	1	18.92	1
6	10.07	1	10.87	1	13.03	1	17.22	1	21.63	1
7	8.494	1	10.14	1	14.08	1	20.13	1	25.43	1
8	8.285	1	10.91	1	16.37	1	23.25	1	28.87	2
9	9.057	1	12.60	1	18.93	1	26.20	2	32.27	2
10	10.33	1	14.77	1	21.50	2	29.25	2	35.77	2
12	14.24	1	19.65	1	27.05	2	35.75	2	43.11	3
14	19.16	1	25.22	2	33.31	3	42.88	3	51.01	3
16	24.94	1	31.57	2	40.31	3	50.71	4	59.57	4
18	31.56	1	38.73	3	48.10	4	59.28	4	68.84	4
20	38.95	1	46.70	3	56.69	4	68.64	4	78.87	5

(1) Configuration 6, TABLE VI

(2) Axial Half-wave Number

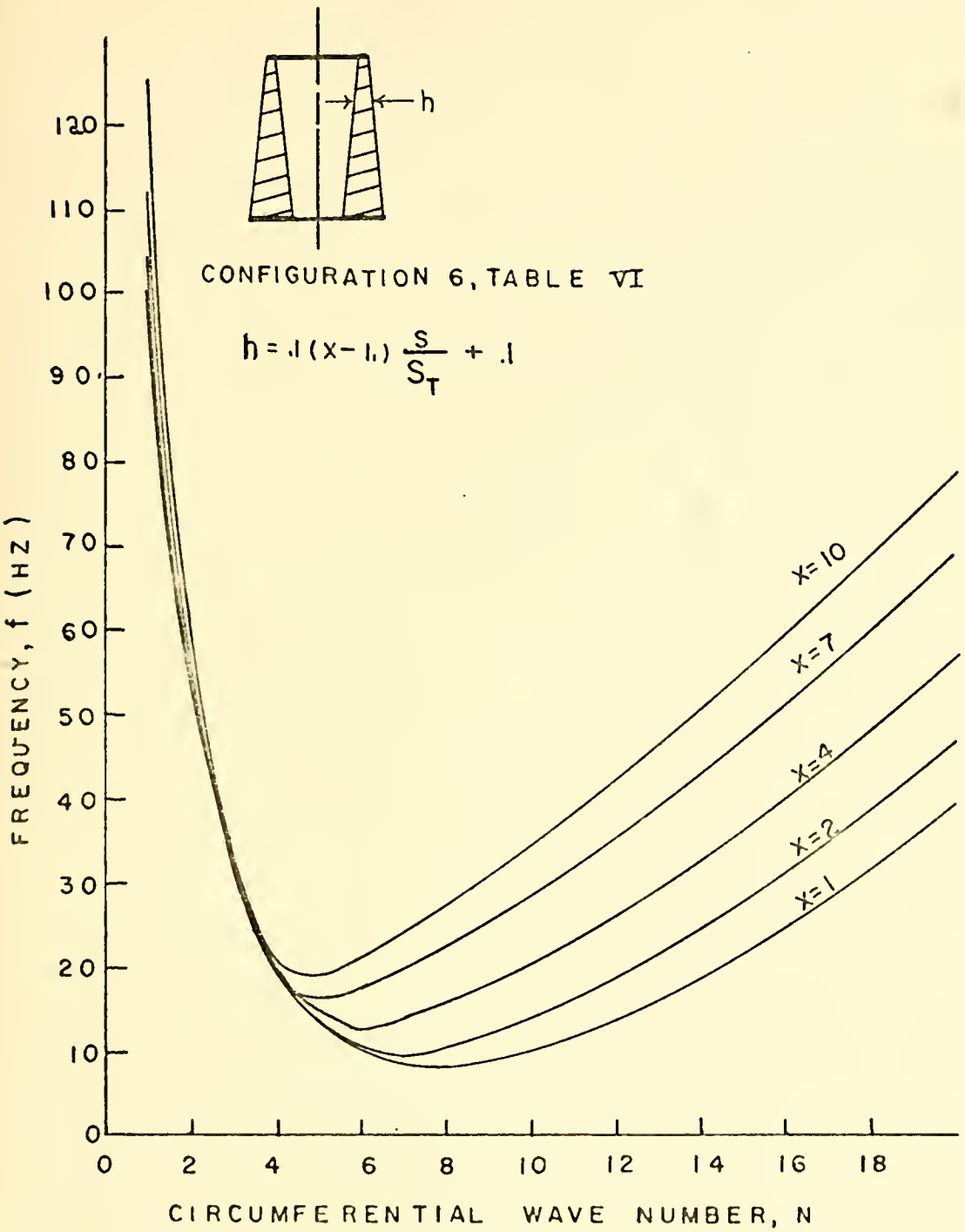


FIGURE 8 EFFECT OF A LINEARLY VARYING THICKNESS ON MINIMUM FREQUENCIES

TABLE VI

SHELL CONFIGURATIONS

Configuration	1	2	3
r (in.)	6.0	100.	A = .5 - s/3 COS(A) - 1.879
$\bar{r}$ (in.)	6.0	100.	1.
r_1(in.)	6.0	100.	.754
r_2(in.)	6.0	100.	.754
R_s(in.)	∞	∞	3.
R_θ(in.)	6.0	100.	r/COS(A)
h (in.)	.015	.1	.001
h_1(in.)	.015	.1	.001
h_2(in.)	.015	.1	.001
ht(in.)	24.0	300.	2.876
S_T(in.)	24.0	300.	3.
$E_y(10^8$ psi)	.3	.3	10^{-8}
ν	.3	.3	.3
$m(10^{-3}$lb-sec^2/in^4)	.732	.733	10^3
BNDRY. COND.(1)			
INITIAL EDGE	FS	FS	FS
FINAL EDGE	FS	FS	FS

(1) Table VII

TABLE VI (Continued)

Configuration	4	5	6
r (in.)	$r_1 + .183S$	$r_1 + .274S$	100.0
$\bar{r}$ (in.)			100.0
r_1 (in.)	3.35	4.81	100.0
r_2 (in.)	5.58	10.9	100.0
R_s (in.)	∞	∞	∞
R_θ (in.)	1.017r	1.04r	100.0
h (in.)	$\frac{.006}{.010}$	.026	$.1\cdot(x-1)\cdot(s/S_T)+.1$
h_1 (in.)	$\frac{.006}{.010}$	.026	.1
h_2 (in.)	$\frac{.006}{.010}$	.026	$.1\cdot x$
ht (in.)	12.0	21.3	300.
S_T (in.)	12.2	22.15	300.
E_y (10^8psi)	.28	.148	.3
ν	.29	.3	.3
m(10^{-3}lb-sec^2/in.^{4})	.740	.4244	.733
BNDRY. COND(1)			
INITIAL EDGE	C	C	FS
FINAL EDGE	F	F	FS

(1) Table VII

TABLE VII

BOUNDARY CONDITION

SPECIFICATION(1)

Case	Specification
Free (F)	$N_s = \hat{N}_{s\theta} = \hat{Q}_s = M_s = 0$
Freely Supported (FS)	$N_s = M_s = 0$ $V = W = 0$
Simply Supported (SS)	$M_s = 0$ $U = V = W = 0$
CLAMPED (C)	$U = V = W = \Phi_s = 0$

(1) The boundary conditions are specified on the total variables N_s, $\hat{N}_{s\theta}$, $\hat{Q}_s$, Φ_s, U, V, W and M_s, as depicted in Figure 2, for all Fourier harmonics considered.

APPENDIX C

CONVERTING SATANS-I TO SATANS-III

PART ONE

PROCEDURE FOR ALTERING COMMON BLOCKS IN TRANSFORMING SATANS-I TO SATANS-III.

1. Remove vector X(4,200) from BL4 and insert in BL104
2. " " Z(4,220) " BL6 " " " "
3. " " ZO(4,220) " BL101 " " " "
4. " " Z2(4,220) " " " " " "
5. " " Z3(4,220) " " " " " "
6. The order of the vectors in common block BL104 is Z, ZDOT, X, ZO, Z2, Z3.
7. Change the order of the vectors in common block BL24 from DL, DG, DF to DG, DL, DF.
8. The order of the remaining vectors in common blocks BL4, BL6 and BL101 should not be altered.

TABLE VIII

CONTENTS OF SATANS-III COMMON BLOCKS

COMMON BLOCK LABEL	CONTENTS
IBL1	MNMAX
IBL2	N(10), MNINIT
IBL3	M0, M1, M2, M3
IBL4	KMAX, KL
IBL5	IBCINL, IBCFNL
IBL6	KLL
IBL7	MNMAXO, MAXD(10), MAXS(10), MAXSY(10), IS(10,10), JS(10,10), ID(10,10), JD(10,10), IJS(10)
or[1]	MNMAXO, MAXD(10), MAXS(10), MAXSY (10), MODVIB(3,100), ITEMP(110)
IBL8	LSTEP, ITR
IBL9	MAXM
IBL10	IFREQ, NTHMAX
IBL11	ICORFL, IPASS
IBL12	KMAX1, KMAX2, NCONV
IBL13	ITRMAX, LSMAX
BL1	A(4,4), BEE(4,4), C(4,4)
BL3	PR(10), PX(10), PT(10)
BL4[2]	P(4,4,200), ZF1M(4,4,10), ZF2M(4,4,10), ZF3M(4,4,10), ZF4M(4,4,10)

TABLE VIII (Continued)

BL5	TT(10), MT(10), DT(10), DMT(10)
BL6[2]	SOE, OSE, ALOAD
BL7	D1, S1
BL8	R(200), GAM(200), OMT(200)
BL9	FFS(4,10), EL1S(4), GEES(4,10)
BL10	PHIX(10), PHIT(10), PHI(10)
BL11	OMXI(200), PHEE, T0, T2
BL12	TDLI, TDEL
BL13	OMEG1(4,4), CAPL1(4,4), OMEGL(4,4), CAPLL(4,4), UNIT(4,4)
BL14	LAM2, LSD18, LSD1N
BL15	NU, U1(10), V1(10), W1(10), V2(10), U2(10), W2(10), U3(10), V3(10), W3(10)
BL16	EPS
BL17	DEL
BL18	EL1(4), ELL(4)
BL19	TH(36)
BL20	DEOMX(200)
BL23	JAY(4,4), H(4,4)
BL24[1][2]	DG(4,4,10), DL(4,4,10), DF(4,4,10)
	or
	DG(4,4,4), RANGE(100,4), TOP, BOTTOM, DELSHF, TOLSHF, RZ(12)
BL25	E(4,4), F(4,4), G(4,4)
BL27	BX3(10), BT3(10), BXT3(10), BE3(10)

TABLE VIII (Continued)

BL28	EXX3(10), ETT3(10), ETX3(10), EXT3(10), EX3(10), ET3(10)
BL29	BX1(10), BT1(10), BXT1(10), BE1(10), BX2(10), BT2(10), BXT2(10), BE2(10)
BL30	EXX1(10), ETT1(10), ETX1(10), EX1(10), ET1(10), EXX2(10), ETT2(10), ETX2(10), EX2(10), ET2(10), EXT2(10)
BL31	DELSQ, EXT1(10)
BL32	TKN, ELAST, CHAR, SIGO
BL34	DEE(4,4,200), DST(4,4,200)
BL100(3)	TEEO, $DYNMC
BL101(2)	DELSD
BL102	DELOAD
BL103	MASS(200)
BL104(1)(2)	Z(4,220), ZDOT(4,220), X(4,200), ZO(4,220), Z2(4,220), Z3(4,220)
or	WV(4,203), XV(4,203), YV(4,203), FILL(28), SHFTPT(3,100)
BL110	TX(10), TTH(10), TXT(10), MX(10), MTH(10), MXT(10), QS(10)
BL111	ABZ, ABZO, ABZN, ABZ3, DD2
BLPLOT	IRADII, IGAMMA, IOMEGS, IOMEGT, IDEOMS, IBSTIF, IDSTIF, IBBSTF, IDDSTF, IPR, IPS, IPT, ITT, IMT, IDTT, IDMT, INS, INTH, INSTH, IQS, IMS, IMTH, IMSTH,

TABLE VIII (Continued)

	IU, IV, IW, IPHIS, IPHIT, IPHI, $PLOTS, $MODAL
BLPLT1	XRADII(100), YGAMMA(100), YOMEGS(100), YOMEGT(100), YDEOMS(100), YBSTIF(100), YDSTIF(100), YBBSTF(100), YDDSTF(100), YPR(100), YPS(100), YPT(100), YTT(100), YMT(100), YDTT(100), YDMT(100), YNS(100), YNTH(100), YNSTH(100), YQS(100), YMS(100), YMTH(100), YMSTH(100), YU(100), YV(100), YW(100), YPHIS(100), YPHIT(100), YPHI(100), XSTATN(100)
BLDATA	TITLE, NO, IMODE, NDIMEN, IPRINT, LCHMAX, IC
BLRUN	IRNAGN
BLK1[4]	VMASS, NN2, $VIBES
BLK2	IVIBES, IVB
BLK3	TWOPI, IBEGIN, IEND, IGO, ISTOP
BLK4	ZERODO, PTFIVE, ONEDO, SMALL, VREAL, VIMAG, AREAL
BLK5	$FLAG, $TAG, $DOVEC
BLK6	NJ, NK, I4, IRGOLD, NUMBER, ITWO
BLK7	SCALEX, SCALEW, SCALE3
BLK8	EPSVIB, EPSVEC, EPSAIT
BLK9	ITR1, ITR2

TABLE VIII (Continued)

BLK10	ASHIFT, ASHFUM, ATSHF, BII, TOLUP, TOLDWN, TOLCLS
BLK11	DIAG(4)
BLK12	NTRY
BLK13	DIF1, DIF2, DIF3, DIF4, TOLLOW, TOLHI, TOLRNG

(1) Identifies a common block of storage that is overlayed by the vibration analysis portion of the program.

(2) Identifies a common block that has been altered in transforming SATANS-I to SATANS-III.

(3) The character $ indicates a LOGICAL * 1 variable.

(4) This and subsequent common blocks designated BLKxx are additions required in transforming SATANS-I to SATANS-III.

(5) Single Precision (REAL*4) variables: JAY, LAM2, LSD18, LSD1N, MASS, MT, NU

(6) Double Precision (REAL*8) variables: EPSVIB, EPSVEC, EPSAIT, ONEDO, PTFIVE, SCALEW, SCALEX, SCALE3, SMALL, TWOPI, VREAL, VIMAG, WV, XV, YV, ZERODO

TABLE IX

SOME FORTRAN VARIABLES

AREAL	single precision form of $-(\omega^r T_o)^2 = +1/\alpha^r$
ASHIFT, ASHFUM, ATSHF	spectral shift points, β
BOTTOM	lower bound on frequency range determined in subroutines RANGER and BTTM
CHAR	characteristic shell reference length, a
DIAG(4)	diagonal of mass matrix $\bar{\mu}$. Location WV(J, 203), J = 1(1)4 stores the double precision form of DIAG(4).
ELAST	E_o
EPSVIB	eigenvalue convergence criterion
EPSVEC	eigenvector convergence criterion
EPSAIT	Aitkin extrapolation convergence criterion
KEY1	input parameter = MODVIB(1,K) for the Kth vibration analysis
KEY2	input parameter = MODVIB(2,K) for the Kth vibration analysis = N(1)
KEY3	input parameter = MODVIB(3,K) for the Kth vibration analysis; determines $DOVEC
MODVIB(3,100)	storage array for input parameters KEY1, KEY2, KEY3
RANGE(100, 4)	storage array for frequency range parameters
SCALEW, SCALEX, SCALE3	α^r, α^{r-1} and α^{r-2} respectively
SIGO	σ_o
SHFTPT(3, 100)	storage array for input values; shift points or frequency range boundaries

TABLE IX (Continued)

TEEO	T_o
TKN	h_o
TOP	upper bound of frequency range determined in subroutine RANGER and TOPPER
VMASS	μ_i
VREAL	double precision form of $-(\omega^r T_o)^2$ $= \frac{+1}{\alpha^r} = \frac{+1.}{SCALEW}$
VIMAG	imaginary part of a complex eigenvalue
WV(4, 203), XV(4, 203), YV(4, 203)	double precision solution vectors for the r, r-1 and r-2 vibration analysis iterations
WT(812), XT(812), YT(812)	singly dimensioned counterparts of WV, XV AND YV.
Z(4, 220)	single precision solution vector
ZV(4, 200)	double precision working vector
$DOVEC	logical operating parameter. If KEY3 = 1, then $DOVEC is true and the vector convergence test will be performed.
$FLAG	logical operating parameter. If $FLAG is true, a real eigenvalue is being dealt with. If $FLAG is false, the program is dealing with a complex eigenvalue.
$TAG	logical operating parameter. If $TAG is true, lower portion of frequency range is being searched. If $TAG is false, upper portion is being searched.
$VIBES	Logical input parameter to designate a free vibration problem.

APPENDIX C

PART TWO

MODIFYING SATNAS-I SUBROUTINES

I. Modify common block statements BL4, BL6, BL24, BL101, and BL104, as specified in Part One, in all affected subroutines. In some cases this means replacing one common block statement with two or vice versa. Hereafter all common blocks are referred to as having been modified. The program is written in FORTRAN and all statements referred to are FORTRAN statements.

II. SUBROUTINE SATANS

1. After card 060 insert the statement

   REAL*8 EPSVIB, EPSVEC, EPSAIT

2. After card 100 insert common block

   IBL7.

3. After card 290 insert common blocks BLK1, BLK2, BLK8, BLK9, and BLK11.

4. Replace the dimension statement, card 300, with the two statements

```
DIMENSION MODVIB(3,100), SHFTPT(3,100), TITLE(18)
EQUIVALENCE (MODVIB(1,1), IS(1,1)), (SHFTPT(1,1),
Z3(1,146))
```

5. Replace cards 350 and 360 with

```
READ(5,101) NO, $DYNMC, $VIBES, IMODE, NDIMEN,
NTHMAX, IFREQ, IPRINT,
1 IBCINL,IBCFNL,KMAX,MNMAX,MAXM,LSMAX,LCHMAX,ITRMAX,IC
```

6. Replace card 490 with the following five cards

```
IF(.NOT.$VIBES) GO TO 10
READ(5,108) IVIBES,DIAG,EPSVEC,ITR2
READ(5,109)(((MODVIB(J,I),J=1,3),(SHFTPT(K,I),
   K=1,3)), I=1,IVIBES)
10 READ(5,107) IRNAGN
   IF($VIBES) CALL INITAA
```

7. After card 650 insert the statement WRITE(6,211)

8. After card 710 insert the statement

```
IF($VIBES) GO TO 11
```

9. Replace card 750 with the following 7 cards

```
   GO TO 6
   card 1180
   card 1190
   GO TO 6
11 WRITE(6,252) KMAX,ITR1,IVIBES,EPS
   WRITE(6,247) CHAR,TKN,ELAST,SIGO,TEEO,NU
 6 IF(.NOT.$PLOTS) GO TO 9
```

10. Replace cards 1170 and 1200 with the statement

```
9 IF(NTHMAX.EQ.0) GO TO 7
```

11. Replace card 1340 with

```
101 FORMAT(I5,L3,L2,14I5)
```

12. Insert the following format statements after card 1400.

```
108 FORMAT(I5,4F1.0,F16.8,I5)
109 FORMAT(3I5,3F15.8)
```

13. Insert the following format statement after card 2250.

```
252 FORMAT(' ',4X,'NUMBER OF STATIONS----------
          ----------',I3,/5X,'M
     1AXIMUM NUMBER OF ITERATIONS----------',
          I3,/5X,'NUMBER OF VIBRATI
     2ON ANALYSES----------',I3,/5X,'CONVERGENCE
          CRITERION(EIGENVALUE
     3S)-----',E12.4//)
```

III. SUBROUTINE DYNAMC

1. Insert common block BLK1 after card 770.
2. Insert the following statements after card 1870.

```
ICHCK2 = ICHCK2 + ICHCK1
IF($PLOTS.AND.(ICHCK2.NE.0)) CALL PLOT1(1)
IF($VIBES) CALL VIBERS(&500)
```

IV. SUBROUTINE PMATRX

1. After card 200 insert the statement

```
IMPLICIT LOGICAL*1 ($)
```

2. After card 340 insert common blocks BL104, BLK1, and BLK10.
3. After card 410 insert the statements

```
ASHFUM = ASHIFT
ATSHF = ASHFUM
```

4. Place card 830 after card 850.
5. After card 840 insert the statement

```
IF($VIBES) GO TO 2.
```

6. Replace card 860 with

```
2 IJ = 1 + KMAX * (MN-1)
```

V. SUBROUTINE EFG

1. Insert common block BLK1 after card 190
2. After card 220 insert the statements

```
VMASS = MAS
IF($VIBES) MAS = 0.0
```

VI. SUBROUTINE ABC

1. After card 050 insert the statement

```
IMPLICIT LOGICAL * 1 ($)
```

2. After card 090 insert common blocks BLK1, BLK10, BLK11 and the two statements

```
DIMENSION TEMP(48)
EQUIVALENCE (TEMP(1),A(1,1))
```

3. Replace card 190, the RETURN statement, with the following series of statements.

```
IF (.NOT.$VIBES) GO TO 4
VMASS = 1.0/(TDEL * VMASS)
DO 2 I = 1, 48
2 TEMP(I) = TEMP(I) * VMASS
DO 3  I = 1,4
BEE(I,I) = BEE(I,I) - ASHFUM*DIAG(I)
3 CONTINUE
4 RETURN
```

VII. SUBROUTINE OUTPUT

1. After card 500 insert common block BL104
2. After card 640 insert common block BLK1
3. After card 670 insert the statement

   IF($VIBES) GO TO 182.
4. After card 890 insert the statement

   IF($VIBES) GO TO 991.
5. Replace card 2320 with

   IF(.NOT.$VIBES) WRITE(6,749)N(MN)

VIII. SUBROUTINE PLOTIT

On cards 080, 090, 100 and 460 change the FORTRAN variable name RANGE to RANGES.

IX. SUBROUTINE DRAWIT

On cards 1860, 1880, 1970, 1980, 1990 and 2000 change the FORTRAN variable name RANGE to RANGES.

APPENDIX C

PART THREE

NEW SUBROUTINES - THEORY AND FLOW CHARTS

Following are descriptions of the subroutines that, added to SATANS-I, constitute the major portion of the conversion from SATANS-I to SATANS-III.

I. SUBROUTINE VIBERS

Subroutine VIBERS is the primary controller for the vibration analysis portion of the program. After the initial input parameters have been printed, subroutine DYNAMC passes control to subroutine VIBERS. Depending on the vibration input parameters subroutine VIBERS calculates and prints the spectral shiftpoint and passes control to the appropriate subroutines. The flowchart for subroutine VIBERS is depicted in Figure 10.

II. SUBROUTINE ITREAL

The purpose of this subroutine is to control the primary iteration process in determining real eigenvalues. The convergence test on the eigenvalues is performed in this subroutine. The vector convergence test is not. That test is performed in subroutine ITRATE. Recall that

$$\alpha^r = \frac{-1}{(\omega^r T_o)^2} = \text{SCALEW}, \quad \text{VREAL} = -(\omega^r T_o)^2$$

and AREAL is the single precision counterpart of VREAL. The flowchart for subroutine ITREAL is depicted in Figure 11.

III. SUBROUTINE ITRATE

In subroutine ITRATE the primary iteration calculations are performed. In addition, if it is to be performed, the vector convergence test is performed in this subroutine.

As discussed previously the primary iteration scheme uses Potters' Gaussian elimination method. For the forward pass

$$ZV_{i,k} = DEE_{i,j,k} * WV_{j,k+1} - DST_{i,j,k} * ZV_{j,k-1}$$

$$\text{for } k = 1(1)KMAX2, \quad i = 1(1)4 \quad \text{and} \quad j = 1(1)4.$$

Similarly for the backward pass

$$WV_{i,k+1} = ZV_{i,k} - P_{i,j,k} * WV_{j,k+2}$$

$$\text{for } k = 1(1)KMAX2, \quad i = 1(1)4 \quad \text{and} \quad j = 1(1)4$$

Separate calculations are used to start and finish the calculations as well as handle shells with initial and/or final poles.

WV is the solution vector for the rth iteration. XV and YV are, respectively, the solutions for the r-1 and r-2 iterations.

WT, XT, and YT are the singly dimensioned versions of, respectively, WV, XV and YV. The flowchart for subroutine ITRATE is depicted in Figure 12.

IV. SUBROUTINE ITCPLX

A. CONVERGENCE FAILURE

A few of the conditions that may cause the inverse iteration scheme to fail are:

1. Eigenvalues may be too close together for separation to occur in the specified maximum number of iterations.
2. A complex conjugate pair of eigenvalues may cause convergence failure.
3. If the two lowest eigenvalues are of equal magnitude and opposite sign, convergence is extremely difficult, if not impossible. This may be caused by shifting. If, for example, the shift point is 6 and three eigenvalues are 4, 6 and 8, the resulting lowest two are plus and minus 2.
4. The occurance of a repeated eigenvalue almost prohibits passing the vector convergence test as do conditions two and three.
5. A poorly conditioned matrix or shifting too close to an eigenvalue may cause a numerically unstable situation in the machine.

The above conditions have not yet occurred when using SATANS-III, but were forced to occur during prototype testing.

B. DESCRIPTION OF SUBROUTINE ITCPLX

If the primary inverse iteration scheme fails to converge, control is passed to subroutine ITCPLX. The method of least squares is used to determine the two eigenvalues closest to the shift point, whether real or a conjugate pair. The eigenvector is not calculated. In the case of the conjugate pair the vector is of no interest (in this application) and in the case of two real eigenvalues a further refinement takes place which automatically includes determination of the two vectors.

C. LEAST SQUARES PROCEDURE

The method of least squares is a procedure for minimizing a function [Refs. 7, 9, 10]. Assume the solution to a problem is found by determining x and y in the following system.

$$f(x,y) = a_1 x + a_2 y - a_3 = 0$$
$$g(x,y) = b_1 x + b_2 y - b_3 = 0 \qquad \text{(C-1)}$$

First form the function $h(x,y) = f^2(x,y) + g^2(x,y)$. Subsequently form the partial derivatives $\frac{\partial h}{\partial x}\Big)_y$ and $\frac{\partial h}{\partial y}\Big)_x$. In the rth step solve for x^r at the minimum with respect to x with the expression resulting from the first

partial derivative. Substitute this x^r in the expression resulting from the second partial derivative and solve for y^r at the minimum with respect to y. Substitute this y^r in the first expression and solve for x^{r+1}. Alternately solve for successive x^r and y^r until they meet some convergence criteria.

D. LEAST SQUARES IN MATRIX NOTATION

System (C-1) may be written as

$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{Bmatrix} x \\ y \end{Bmatrix} - \begin{Bmatrix} a_3 \\ b_3 \end{Bmatrix} = 0$$

and in general as

$$[a_{ij}] \{u_j\} - \{v_i\} = 0 \qquad \begin{matrix} i = 1(1)m \\ j = 1(1)n \end{matrix} \qquad \text{(C-2)}$$

or as

$$A\,U - V = 0 \qquad \text{where m need not equal n.}$$

The least squares procedure may then be performed by forming

$$A^T A U - A^T V = 0 \qquad \text{and solving for the } u_j.$$

By way of demonstration start with (C-2) and perform the matrix multiplication and rewrite as

$$\left\{ \sum_{j=1}^{n} a_{ij}u_j - v_i \right\} = 0 \quad i = 1(1)m$$

Form the function

$$h = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} a_{ij}u_j - v_i \right)^2$$

and the partial derivatives

$$\left.\frac{\partial h}{\partial u_k}\right)_{\substack{u_j \\ j\neq k}} = 2 \sum_{i=1}^{m} a_{ik} \left(\sum_{j=1}^{n} a_{ij}u_j - v_i \right) = 0$$

for $k = 1(1)n$.

Delete the 2 as immaterial to the minimizing procedure and write the $\left.\frac{\partial h}{\partial u_k}\right)_{\substack{u_j \\ j\neq k}}$ as a system. Note that the row vector

$$\lfloor a_{ik} \rfloor = \{a_{ij}\}^T \qquad \text{for } j = k.$$

$$\sum_{i=1}^{m} a_{ik} \left(\sum_{j=1}^{n} a_{ij}u_j - v_i \right)$$

$$= [a_{ik}] \left(\sum_{j=1}^{n} a_{ij}u_j - v_i \right) = 0$$

$$[a_{ik}] \left([a_{ij}]\{u_j\} - \{v_i\} \right) = 0$$

$$[a_{ij}]^T [a_{ij}] \{u_j\} - [a_{ij}]^T \{v_i\} = 0$$

$$A^T A U - A^T V = 0$$

E. SUBROUTINE PROCEDURE

Assume that the two eigenvalues closest to the shift point are the roots of the quadratic equation $x^2 - ax - b = 0$. ITCPLX determines a and b from which the roots may be extracted.

The equation is true for any vector $Z = \{z\}$.

$$(x^2 - ax - b)\, Z = x^2 Z - ax Z - b Z = 0$$

From the iteration process on the rth iteration

$$xZ = \alpha^{r+1} Z^{r+1} \quad \text{where } \alpha^{r+1} \text{ is the maximum}$$

element in Z^{r+1} before scaling. It then follows that

$$\alpha^{r+2}\ \alpha^{r+1}\ Z^{r+2} - a\ \alpha^{r+1} Z^{r+1} - b\ Z^{r} = 0$$

Rewriting the above

$$a\ \alpha^{r+1} Z^{r+1} + b\ Z^{r} = \alpha^{r+2}\ \alpha^{r+1}\ Z^{r+2}$$

and

$$[Z] \begin{Bmatrix} \alpha^{r+1} & a \\ & b \end{Bmatrix} = \alpha^{r+2}\ \alpha^{r+1}\ Z^{r+2}$$

where

$$[Z] = [Z^{r+1}\ Z^{r}]$$

Performing the least squares procedure

$$[Z]^T\ [Z] \left\{ \begin{matrix} \alpha^{r+1} & a \\ & b \end{matrix} \right\} = \alpha^{r+2}\ \alpha^{r+1}\ [Z]^T\ \{Z\}^{r+2}$$

Moving the α^{r+1} to the first column and rewriting

$$\begin{bmatrix} \alpha^{r+1} & Z^{r+1^T} Z^{r+1} & & Z^{r+1^T} Z^{r} \\ \alpha^{r+1} & Z^{r^T} Z^{r+1} & & Z^{r^T} Z^{r} \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix}$$

$$= \alpha^{r+1}\ \alpha^{r+2} \begin{Bmatrix} Z^{r+1^T} & Z^{r+2} \\ Z^{r^T} & Z^{r+2} \end{Bmatrix}$$

The α^{r+1} and Z^{r+1} are solved for by Potters' Gaussian elimination procedure. The two eigenvalues are then $s + t$ and $s - t$ where $s = \frac{a}{2}$ and $t = \frac{\sqrt{a^2 + 4b}}{2}$. The test for two real roots versus a complex conjugate pair is the sign of the discriminant $a^2 + 4b$. The two convergence tests are

$$\left| \frac{|s^{r+1}| - |s^r|}{s^{r+1}} \right| \quad \text{less than EPSVIB}$$

and

$$\left| \frac{|t^{r+1}| - |t^r|}{t^{r+1}} \right| \quad \text{less than EPSVIB.}$$

F. SUBROUTINE FORTRAN VARIABLES

In subroutine IPCPLX the Fortran Variables are

SCALEW	$= \alpha^{r+2}$	WT(I)	$= Z^{r+2}$
SCALEX	$= \alpha^{r+1}$	XT(I)	$= Z^{r+1}$
SCALE3	$= \alpha^{r}$	YT(I)	$= Z^{r}$
ZA	$= a$	ZB	$= b$
VREAL	$= s^{r+1}$	VIMAG	$= t^{r+1}$
OLDVR	$= s^{r}$	OLDVI	$= t^{r}$

The system set-up is

$$\begin{bmatrix} B & C \\ E & F \end{bmatrix} \begin{Bmatrix} ZA \\ ZB \end{Bmatrix} = \begin{Bmatrix} D \\ G \end{Bmatrix}$$

where

$$B = SCALEX * \left(\sum_{I=IGO}^{ISTOP} XT(I) * YT(I) \right)$$

and

$$G = SCALEX * SCALEW * \left(\sum_{I=IGO}^{ISTOP} YT(I) * WT(I) \right)$$

etc.....

and

```
VREAL = 0.5 * ZA
VIMAG = 0.5 * DSQRT (ZA*ZA + 4.0 * ZB)
```

G. PERFORMANCE OF SUBROUTINE ITCPLX

As mentioned before, ITCPLX was never invoked by SATANS-III during the execution of the test cases. However, during prototype testing it was possible to evaluate the procedure extensively [Refs. 7, 8, 16 and 17]. In all cases, whenever the inverse iteration scheme failed subroutine ITCPLX returned the correct eigenvalues, both real and complex. This subroutine provides an excellent backup and diagnostic tool for the basic solution procedure.

Since the subroutine uses Potters' scheme to determine the Z^{r+1} it is subject to the same general ill-conditioning problem that the primary scheme is subject to in this respect. A second problem arises when t^r is extremely small. This is a highly unlikely case in practice in regards to this application. In the case of total non-convergence by both schemes the results of the last two iterations of both schemes are printed along with the information on whether they would be real or complex values. This information is valuable in analyzing a difficult problem. The flowchart for subroutine ITCPLX is depicted in Figure 13.

V. SUBROUTINE RANGER

A. FREQUENCY RANGE SEARCH

The primary objective in a frequency range search operation is to provide some method of ensuring that all the frequencies in the given range are determined or, if missed, determining which are missed. This is accomplished

in two ways in SATANS-III. Each eigenvalue is determined three times in covering a range. For each eigenvalue determined, the mode shape is simultaneously determined and printed once, along with the eigenvalue (frequency). Determining an eigenvalue three times is time consuming but reduces the chance of missing a mode and examining the mode shapes determined is an easy way of spotting a missed mode.

The method employed uses the basic iteration scheme to actually determine the eigenvalues. No a priori knowledge of the number or distribution of frequencies in the range is assumed. The amount of work involved is directly proportional to the number, separation and distribution of frequencies in the range and the operating parameters.

The calculations involve two primary subroutines, RANGER and ADJUST. Subroutine RANGER contains the primary search scheme and initial eigenvalue determination scheme. Two other subroutines are involved, BTTM and TOPPER, which determine the lowest and highest frequencies in the range, respectively.

The only additional inputs by the user for this operation are the frequency range endpoints in Hz.

B. SEARCH SCHEME

The eigenvalues naturally determined by the system of four second order equations are negative ($-\omega^2 T_o^2$). It is easier to search the field of negative numbers rather than alter the set-up of the problem to yield positive numbers. For this reason the lower bound of the range,

BOTTOM, is the negative number smallest in magnitude and corresponds to the lowest frequency. Similarly the upper bound of the range, TOP, is the negative number largest in magnitude and corresponds to the highest frequency.

For example if the range from 1.59 Hz to 5.29 Hz is to be searched then BOTTOM = $-(6.28(1.59)1.0)^2 = -100$ where 6.28 = 2 PI and 1.0 = T_o and TOP = $-(6.28(5.29)1.0)^2$ = - 1,100. These two numbers establish the initial range. This initial range is subsequently partitioned into a number of smaller ranges as new eigenvalues are determined, and each smaller range is eliminated when all eigenvalues within that range have been determined.

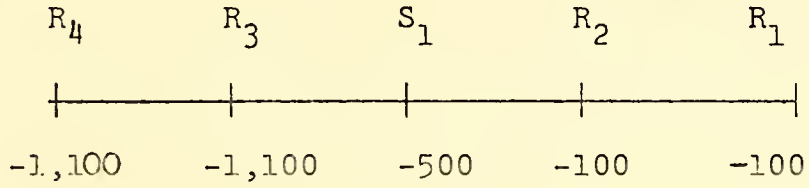
Each range has associated with it four parameters stored in the array RANGE (100,4). For the sake of clarity the elements of this array will be referred to as R_i in the example that follows. For a given range, K, the elements of the array RANGE are

	EXAMPLE
1. Lowest eigenvalue in the range, RANGE(K,1)	$R_1 = -100.$
2. Either the lowest eigenvalue or the maximum value in the bandwidth from which it was determined. RANGE(K,2)	$R_2 = -100.$

3. Either the highest eigenvalue or the minimum value in the bandwidth from which it was determined. RANGE(K,3) — R_3 = - 1,100.

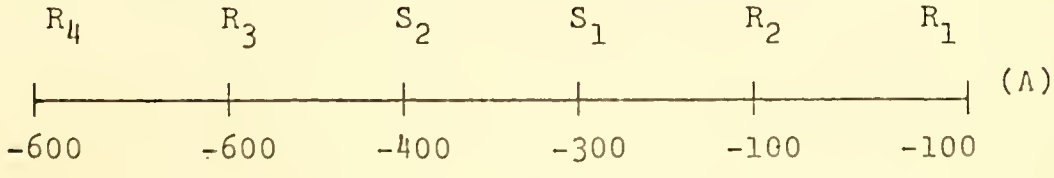
4. Highest eigenvalue in the range. RANGE(K,4) — R_4 = - 1,100.

Therefore, RANGE(K,J) for J = 1(1)4 is (-100., -100., - 1,100., - 1,100.), and is represented graphically as

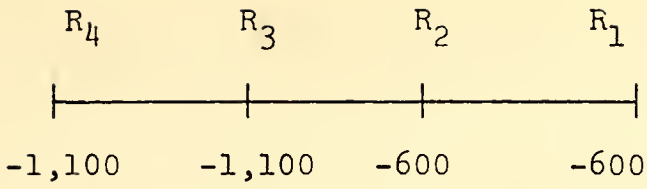


The range of frequencies is searched by choosing a suitable spectral shift point. (Present program parameters call for 40% up from the bottom of the range specified by R_2. This may be a user specified parameter by altering the statement TOLCLS = .4 in subroutine INITAA). In the example the first shift point is -500, S_1.

Assume a new eigenvalue, -600, is determined. The initial range is then partitioned in two ranges as shown below.



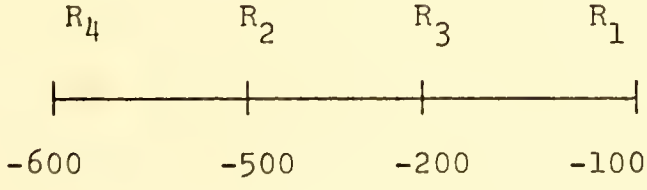
(A)



(B)

The above partitioning occurs each time a new eigenvalue is determined. The next shift point, S_1, is -300.

Assume that there are no more eigenvalues in the range from -100 to -600. It is the nature of the inverse iteration process that the eigenvalue closest to the spectral shift point will be determined. This is -100 in the example. Because of this property it is assumed that no other eigenvalue exists within ±200 of S_1. (Otherwise it should have been determined.) If the eigenvalue -100 is determined from the shift point -300 then R_2 is set to the maximum value of the bandwidth. $R_2 = -300 + (-300 - (-100)) = -500$. The next shift point is $S_2 = -400$ (40% down from the top.). From the spectral shift point the higher eigenvalue, -600, will be determined and a similar calculation for R_3 is performed; $R_3 = -400 - (-600 - (-400)) = -200$. The following situation then exists

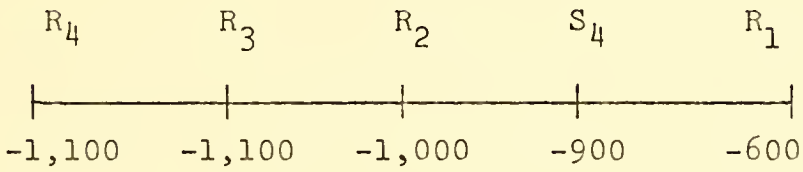


This is how a range is eliminated as being completely covered. The range elimination test is

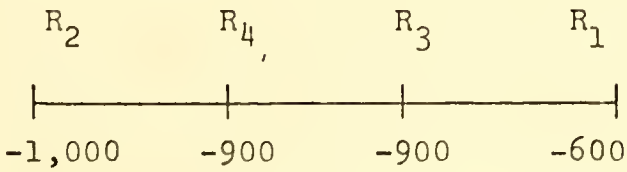
$$\text{Is } R_2 = < \text{TOLRNG} * R_3 \text{ ?} \quad (\text{TOLRNG} = 1.0 - (.4*\text{EPS}))$$

and is intended to prevent missing a mode near the center of the range when a shift point of 25% from the top or bottom is used. (TOLRNG is set in subroutine INITAA.)

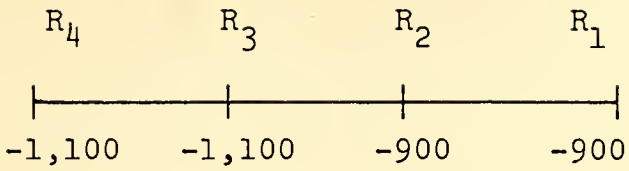
The first shift point, S_3, in the second range, (B), is -800, and assume further that the lower eigenvalue of -600 has been determined (for the third time !!!). The situation is as follows



The next shift point (40% down from the top as specified by R_3) is -900 = S_4. From this spectral shift point assume that an eigenvalue of -900 is determined. The partition would then be



(B1)



(B2)

with range (B1) being immediately eliminated. The process would be repeated with range (B2).

A range, or portions thereof, may be eliminated because no real eigenvalue is found within the range, or because an eigenvalue outside the range is found. Consideration must also be given to the situation where ill-conditioning would cause the primary scheme to fail and two real eigenvalues might be determined by subroutine ITCPLX.

In deciding if an eigenvalue has been previously determined when eliminating a range, consideration should be given to the desired results of the problem. For example, is it necessary to know all frequencies in a range even if four of them differ by less than 1%? A set of parameters (TOLLOW and TOLHI) determine the degree of separation that defines individual eigenvalues and are presently set at ± 1/2%. They are set in subroutine INITAA. Thus, if two eigenvalues differ by less than ± 1/2% they are defined as one eigenvalue.

Subroutine RANGER has incorporated an Aitkin extrapolation to accelerate the convergence rate of the eigenvalues. To further reduce computation time the convergence criteria, EPSAIT, on the accelerated

eigenvalues is different from the criteria, EPSVIB, for the refined eigenvalues. EPSAIT should be commensurate with the chosen separation criteria. EPSAIT is currently set at .01 in subroutine INITAA.

Since the lowest frequencies are of most interest the search scheme is biased toward the lower frequencies. The shift points chosen first are in the lower portion of the range. Second is the choice of vectors. For the first vector the initial guess is all ones. Subsequently the guess is the difference between the rth and r-2 iterated vector. The iterated vectors are richer in the lower modes than the higher modes. (Consider the expansion of an arbitrary vector in the set of othogonal eigenv ectors.) This amounts to a pseudo sweeping technique. The actual calculation of the difference of the two vectors is performed in subroutine ITRATE. Figure 9 depicts the search scheme schematic. The flowchart for subroutine RANGER is depicted in Figure 14.

VI. SUBROUTINE ADJUST

Subroutine ADJUST is the second of the subroutines involved in the frequency range search operation. (The first was subroutine RANGER.) Three main functions are performed by this subroutine.

A. Newly determined eigenvalues and eigenvectors are refined.

B. Range partition and elimination calculations are

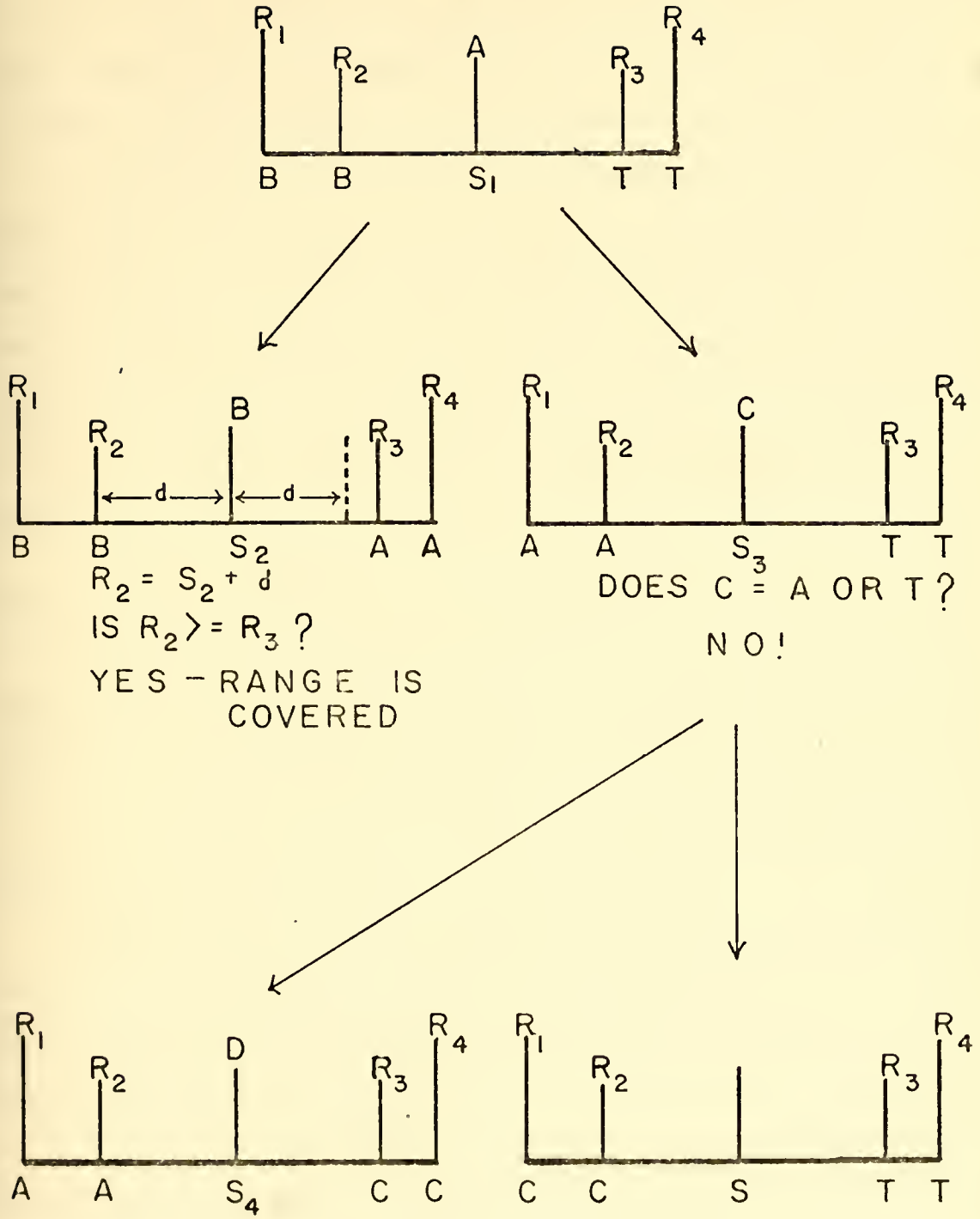


FIGURE 9 SEARCH SCHEME SCHEMATIC

performed including the range elimination test.

C. The situation of a complex conjugate pair or two real eigenvalues, both in the range, only one in the range or both outside the range, is handled in this subroutine.

In refining a newly determined eigenvalue, a new shift point close to the subject eigenvalue is chosen. How close is determined by the parameter TOLSHF (currently set at 90%) set in subroutine INITAA. A new vector is not obtained and the iteration begins with the last obtained vector. Figure 15 depicts the flowchart for subroutine ADJUST.

VII. SUBROUTINE BTTM

Subroutine BTTM is one of two minor subroutines (BTTM and TOPPER) called in subroutine RANGER. The function of BTTM is to determine the minimum eigenvalue of the range. The initial shift point used is the lower bound specified by the user in his input statement.

If a real eigenvalue cannot be found at the first shift point a new shift point 10% up the range is tried. This continues until the range is completely covered or a real eigenvalue is determined.

A form of the Aitkin extrapolation is used to accelerate the initial convergence rate.

The vector convergence test is always performed in this subroutine in determining the minimum eigenvalue. The flowchart for subroutine BTTM is depicted in Figure 16.

VIII. SUBROUTINE TOPPER

Subroutine TOPPER operates in a manner analogous to subroutine BTTM in determining the maximum eigenvalue of the range.

The main differences are the initial shift point, which is the upper bound specified by the user, and the fact that the vector convergence test is not performed unless specified by the user. The flowchart for subroutine TOPPER is depicted in Figure 17.

IX. SUBROUTINE INITAA

Subroutine INITAA, called only in a vibration analysis, sets the operating parameters of the program.

IBEGIN, IEND, IGO and ISTOP are do-loop begin and terminate parameters to ease handling of the eigenvectors as doubly and singly dimensional arrays.

The rest of the program is self-explanatory. See the program listing. Figure 18 depicts the flowchart for subroutine INITAA.

X. SUBROUTINE START

Subroutine START initializes iteration parameters and the starting vector. The flowchart for subroutine START is depicted in Figure 19.

XI. SUBROUTINE VECOUT

This subroutine controls the printed output for the vibration analyses. Several operations are performed in

addition to printing output.

A. The eigenvalues are converted to real time cycles per second (Hz).

B. A check is made for imaginary frequencies.

C. The double precision solution vector WV(I) is mapped into single precision space prior to calling subroutine OUTPUT which prints the solution vector.

Examples of all possible printed messages are presented in Figure 20. Lines in braces are optional or alternate possibilities.

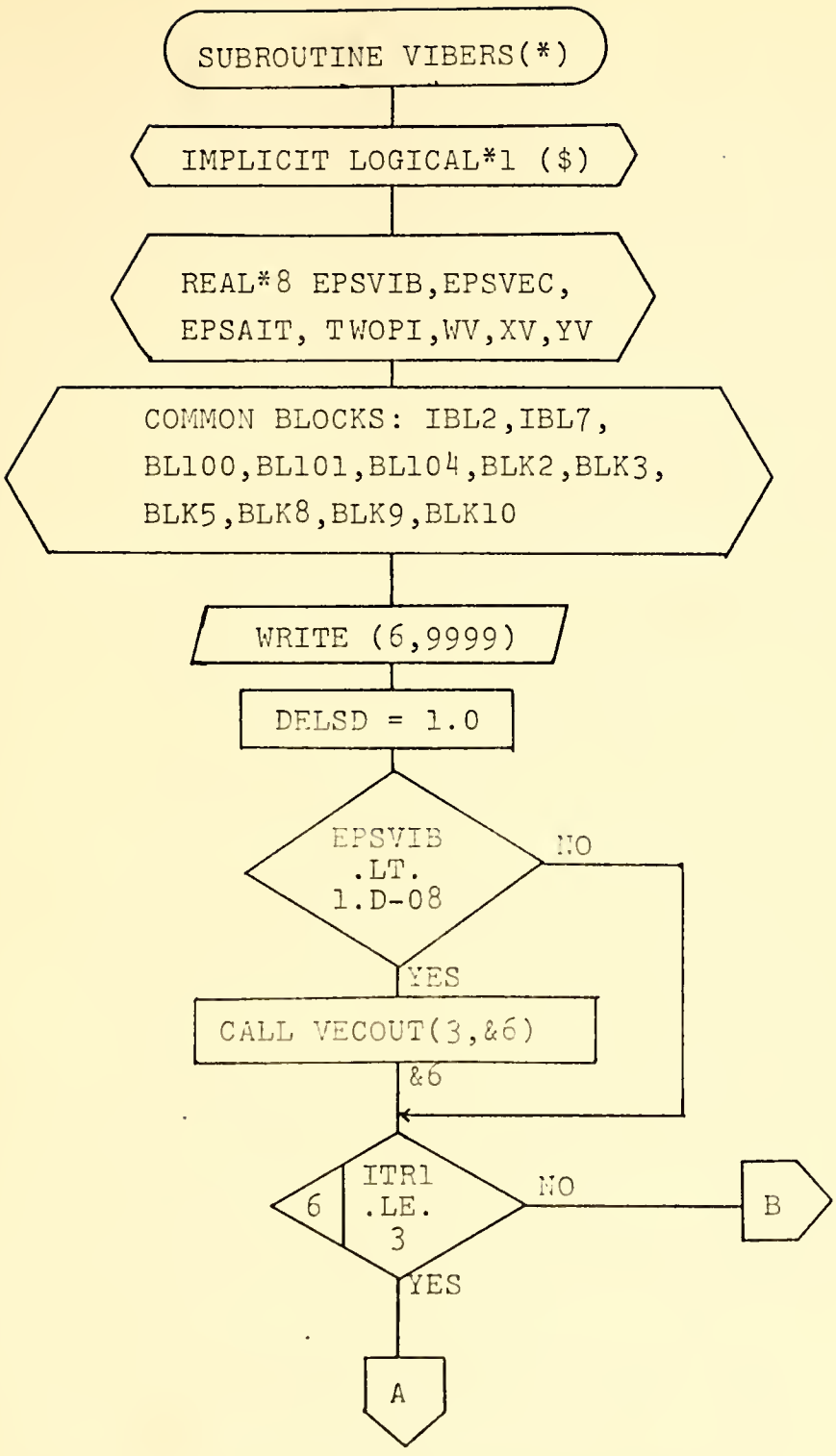


FIGURE 10. SUBROUTINE VIBERS(*) FLOWCHART

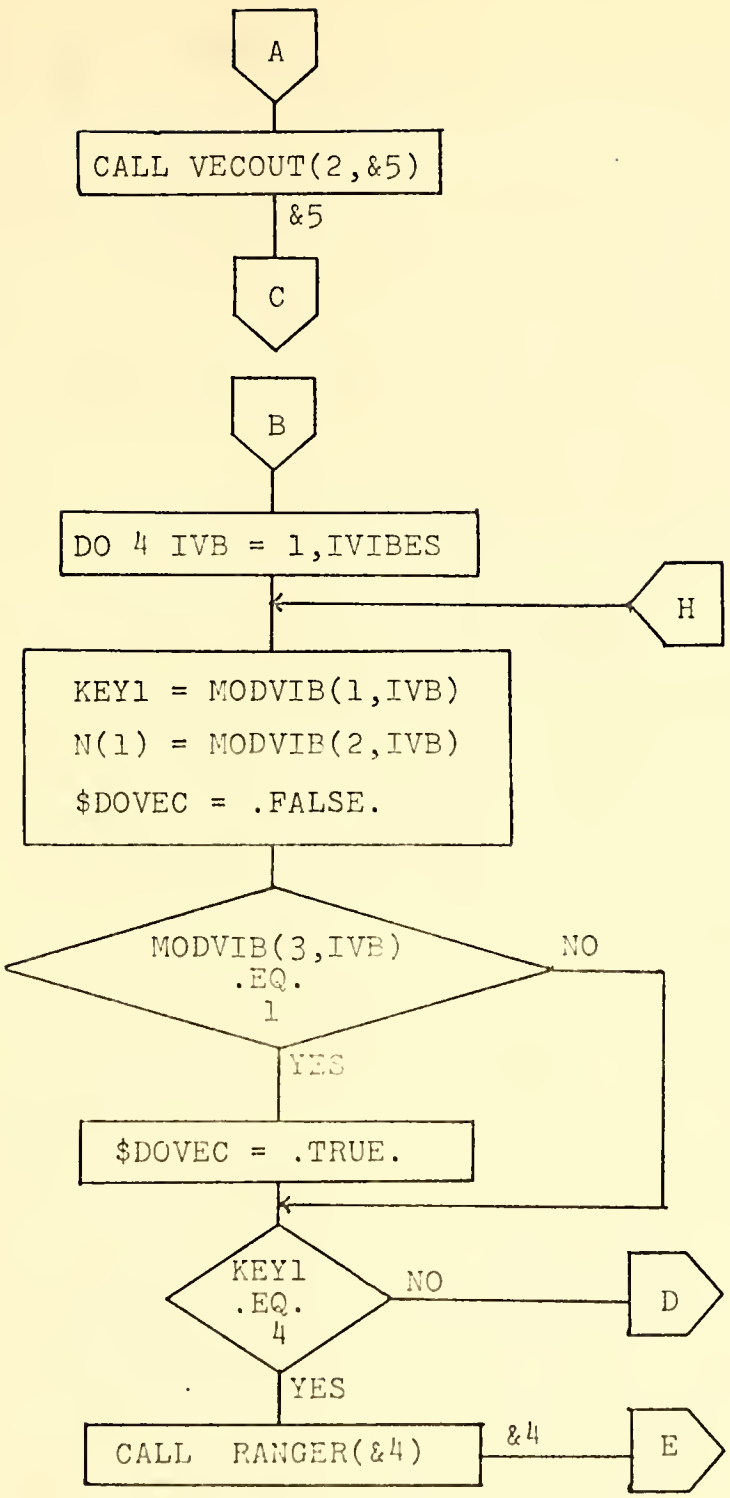


FIGURE 10 continued

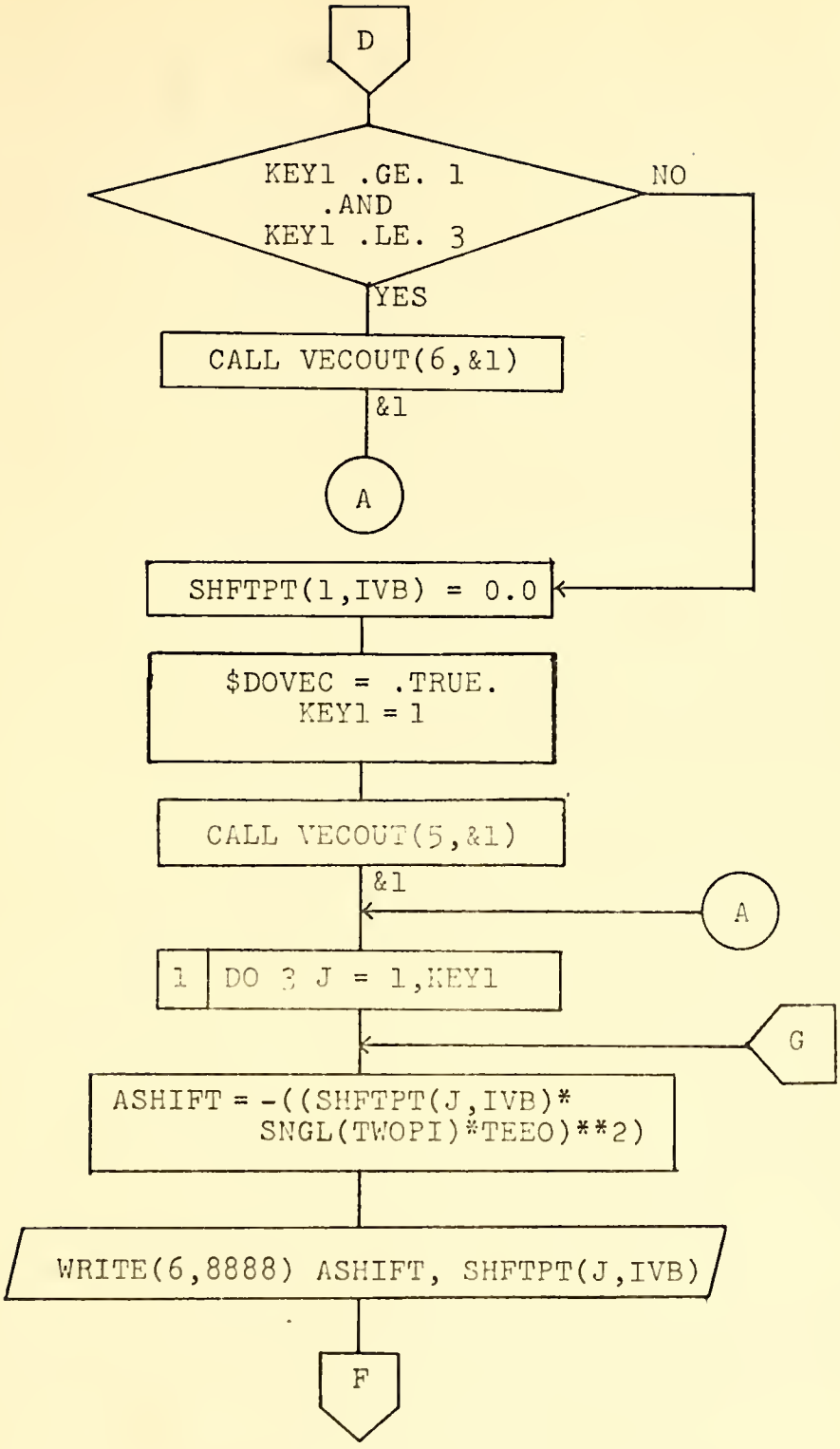


FIGURE 10 continued

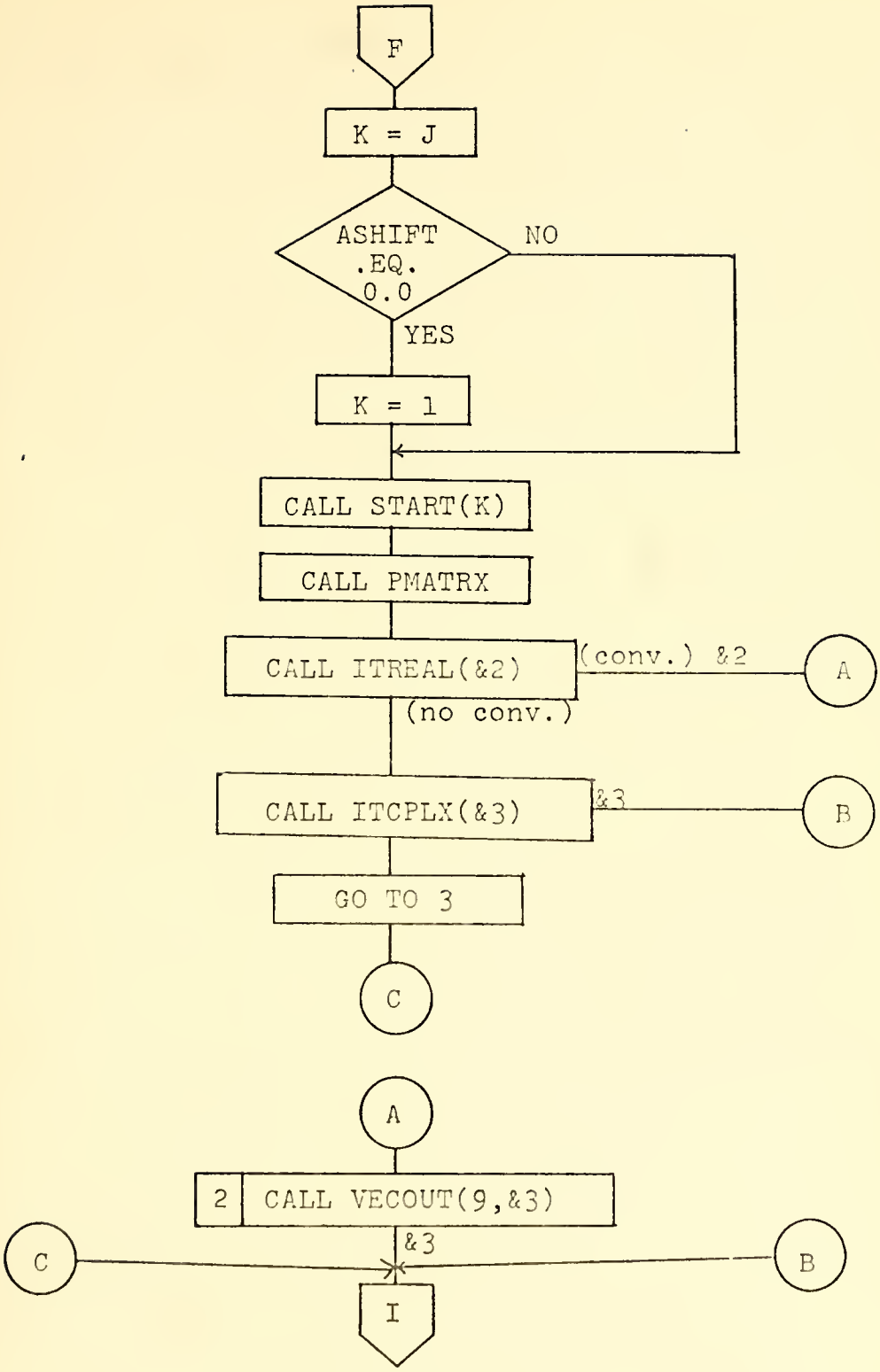


FIGURE 10 continued

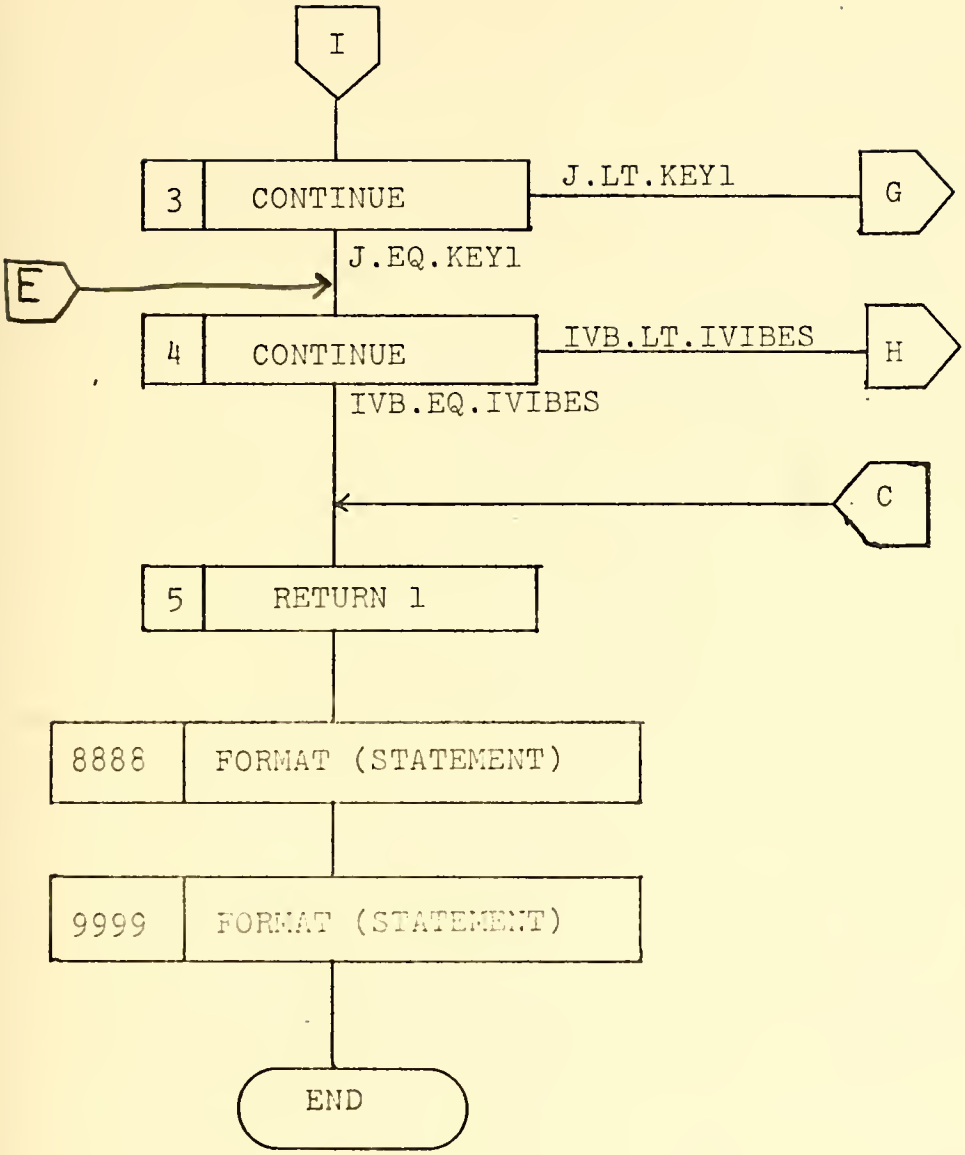


FIGURE 10 continued

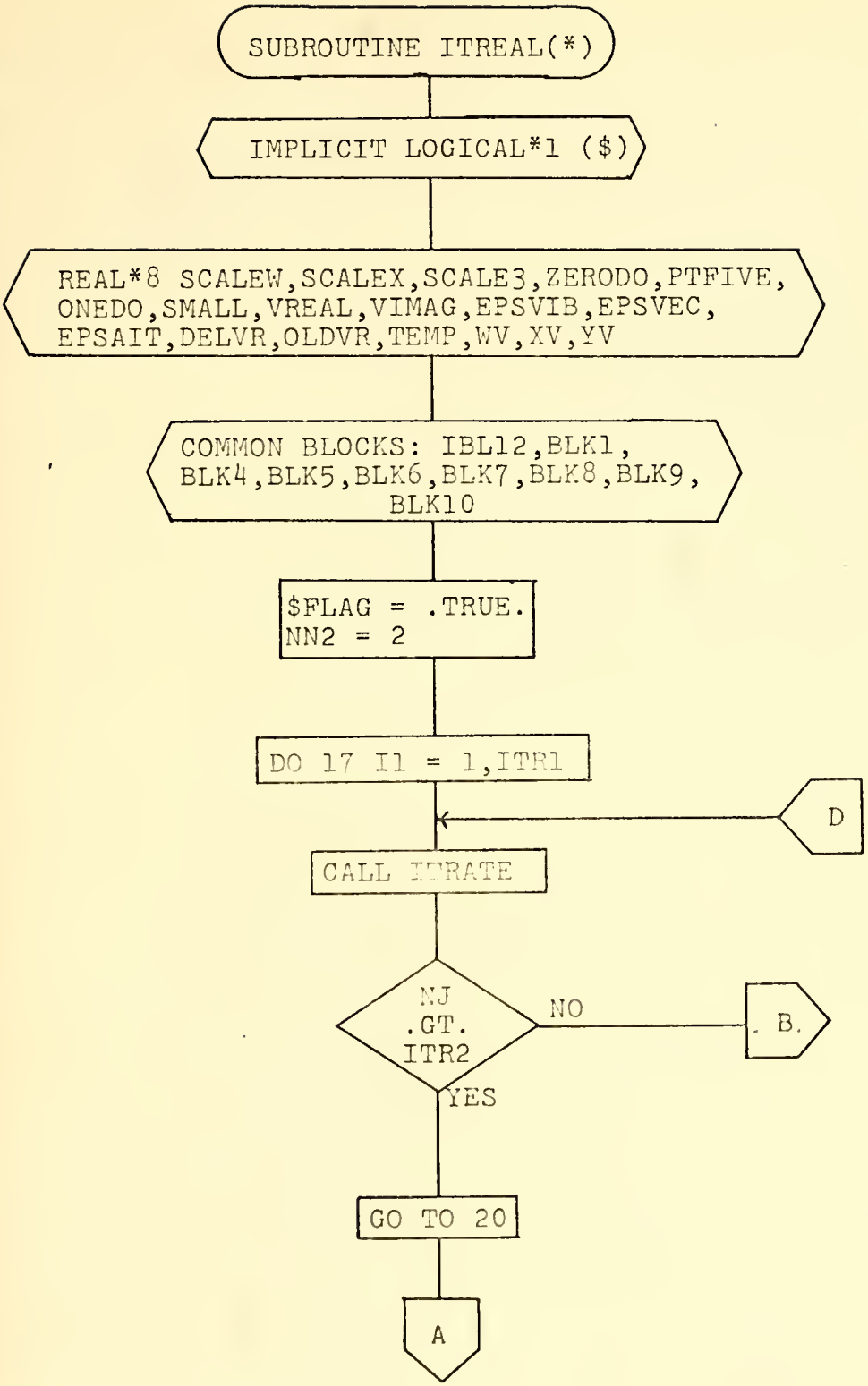


FIGURE 11. SUBROUTINE ITREAL(*) FLOWCHART

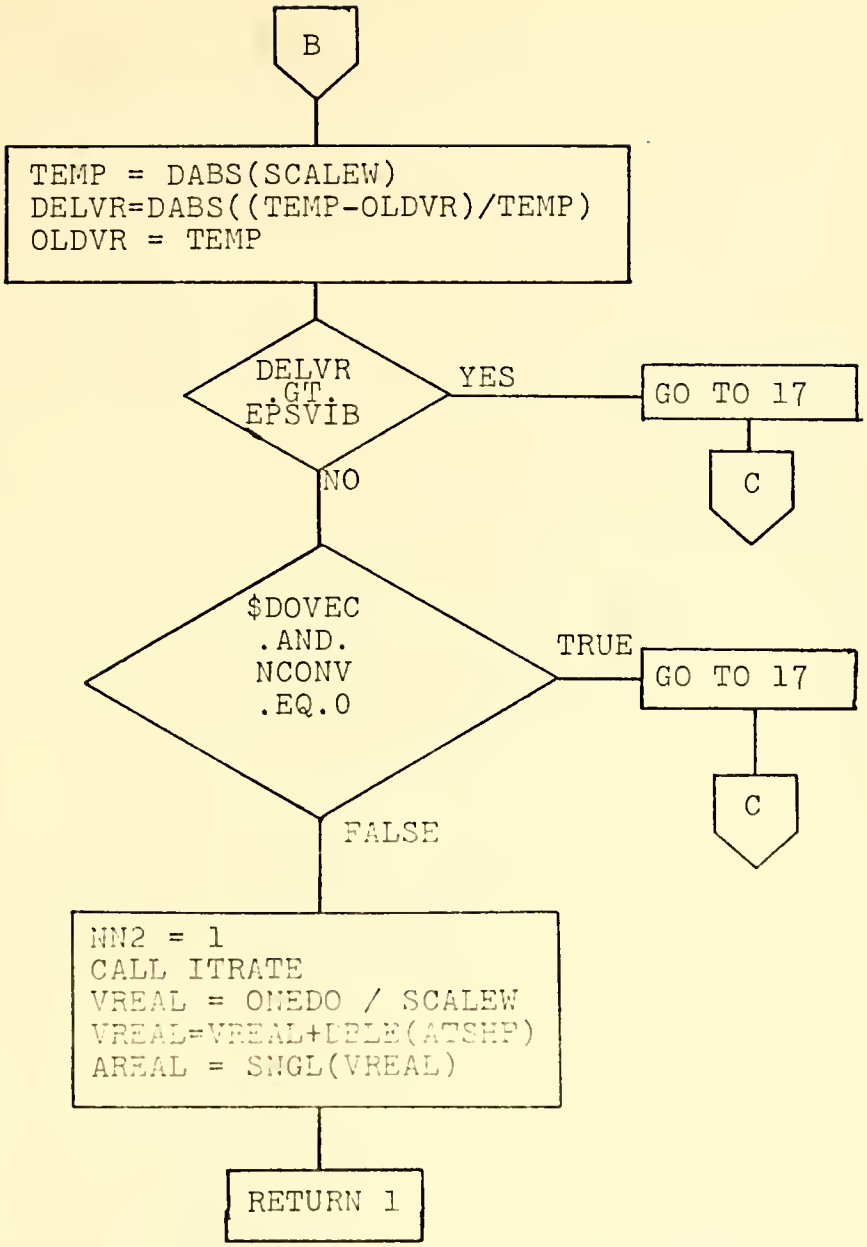


FIGURE 11 continued

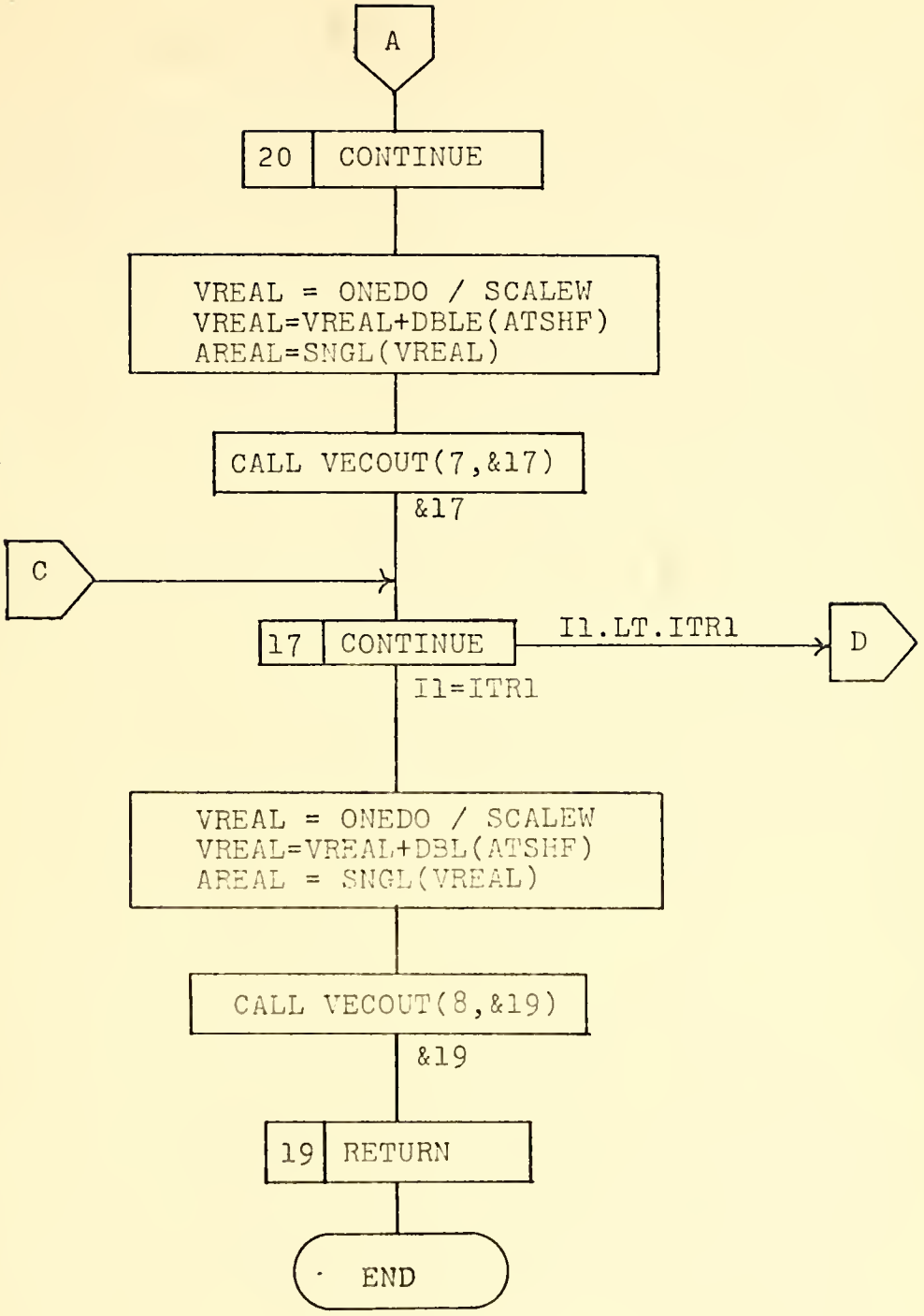


FIGURE 11 continued

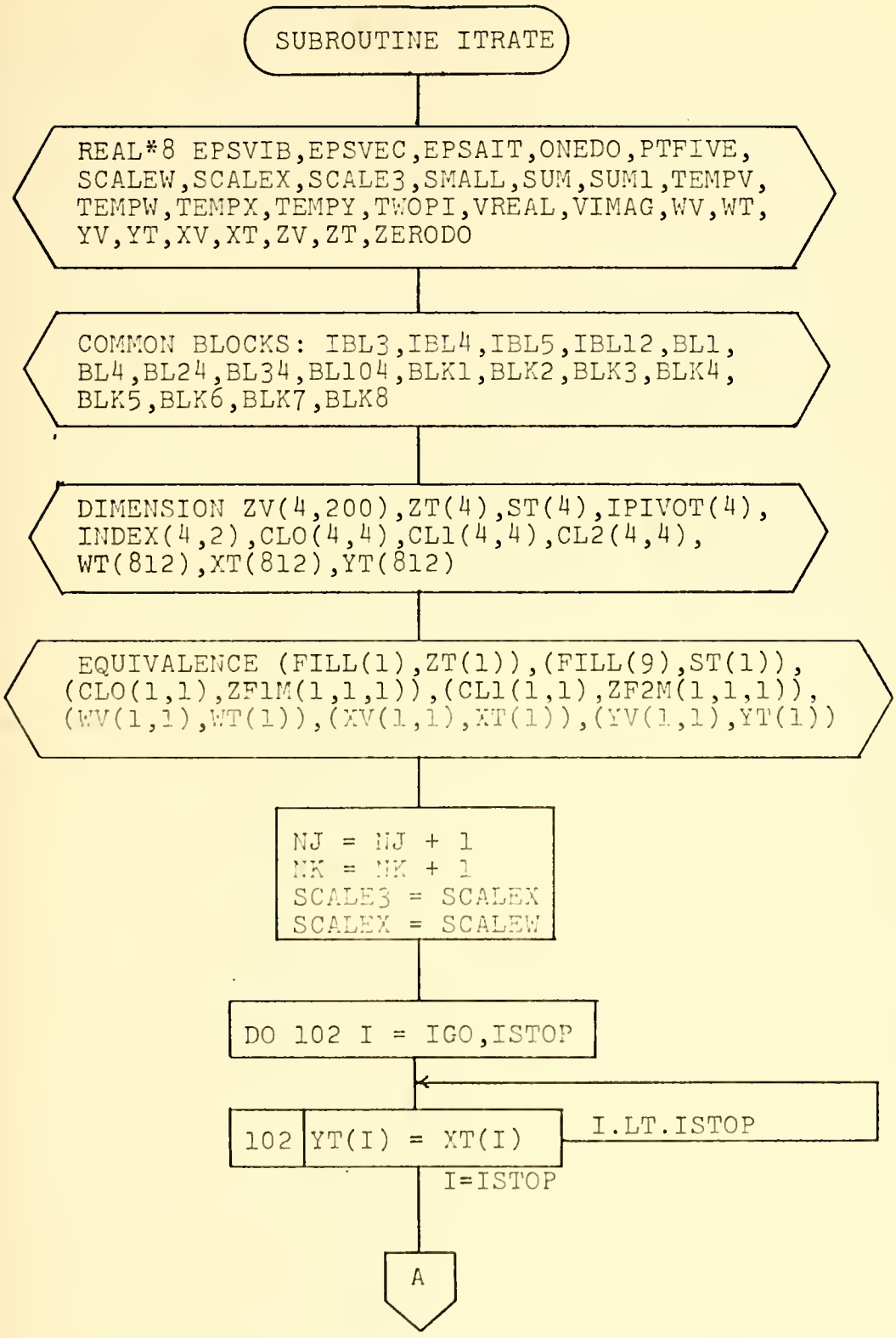


FIGURE 12. SUBROUTINE ITRATE FLOWCHART

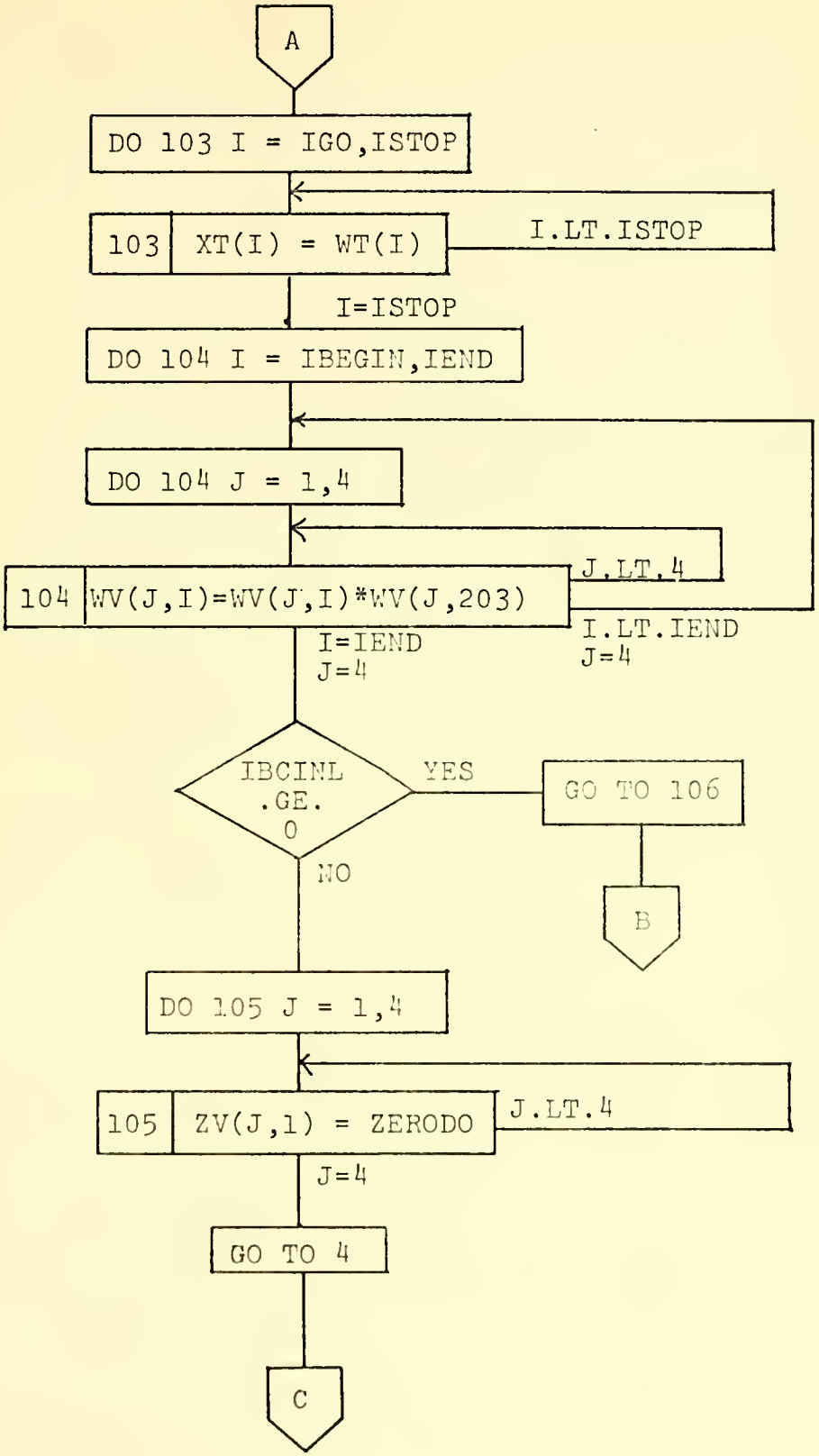


FIGURE 12 continued

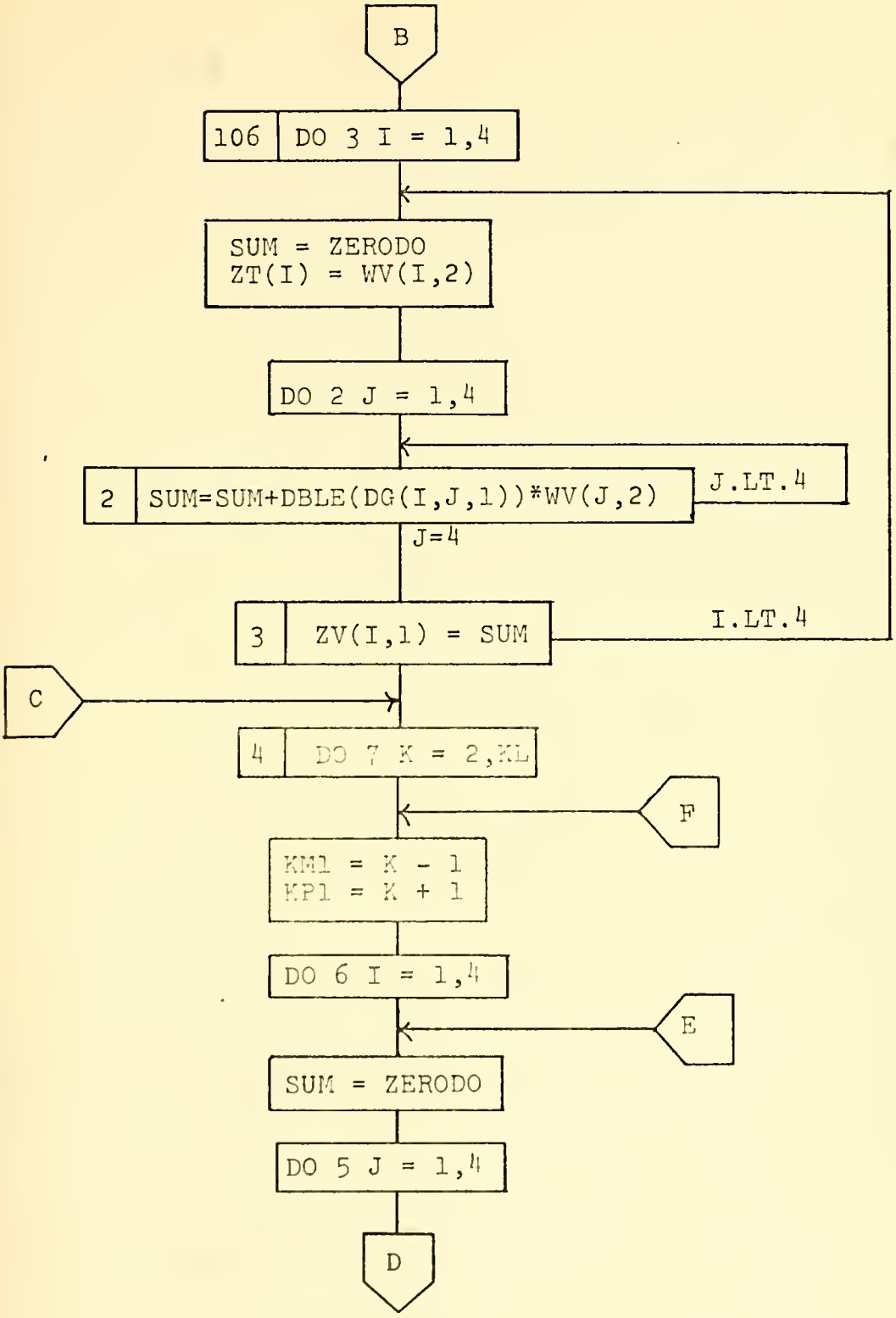


FIGURE 12 continued

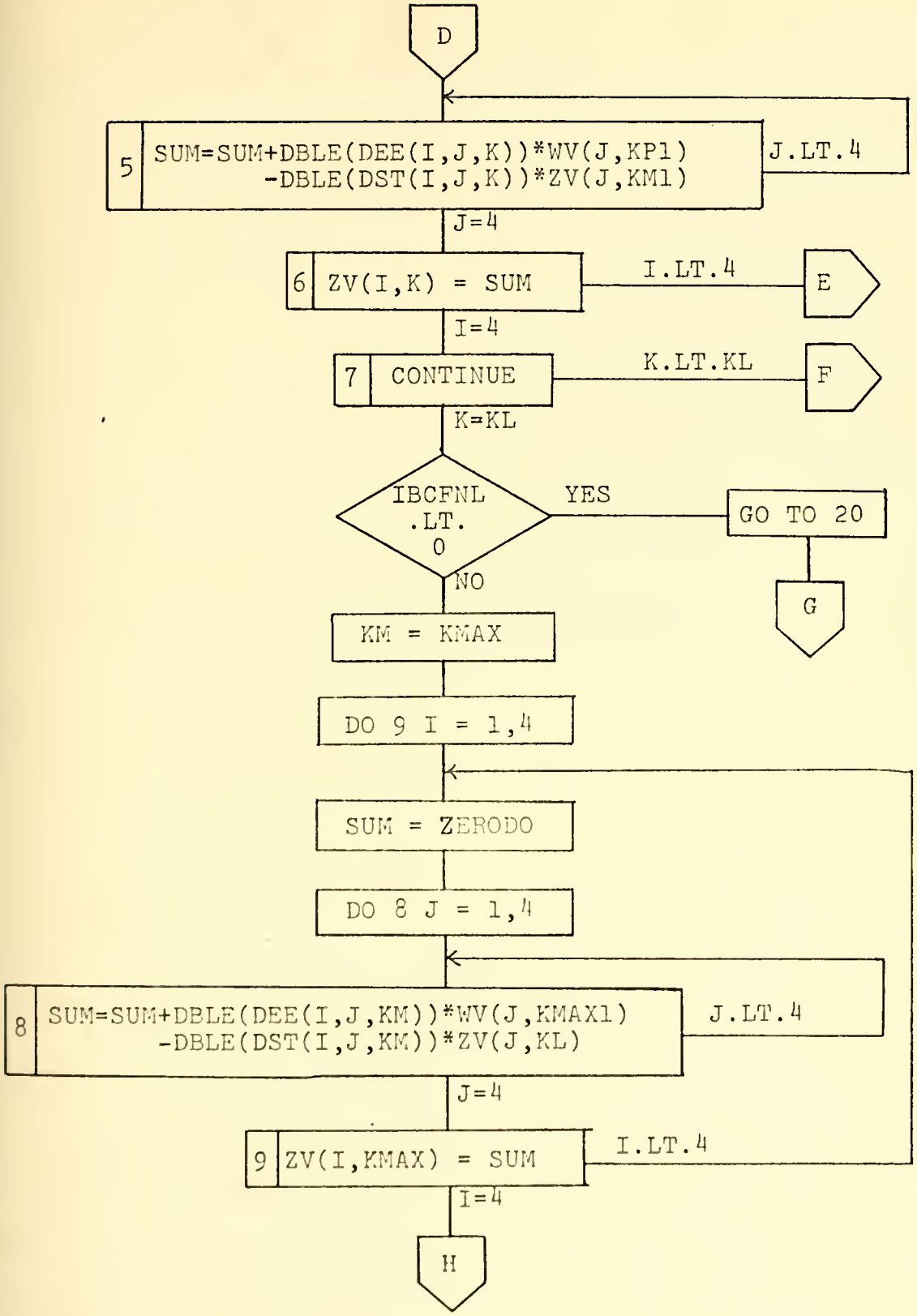


FIGURE 12 continued

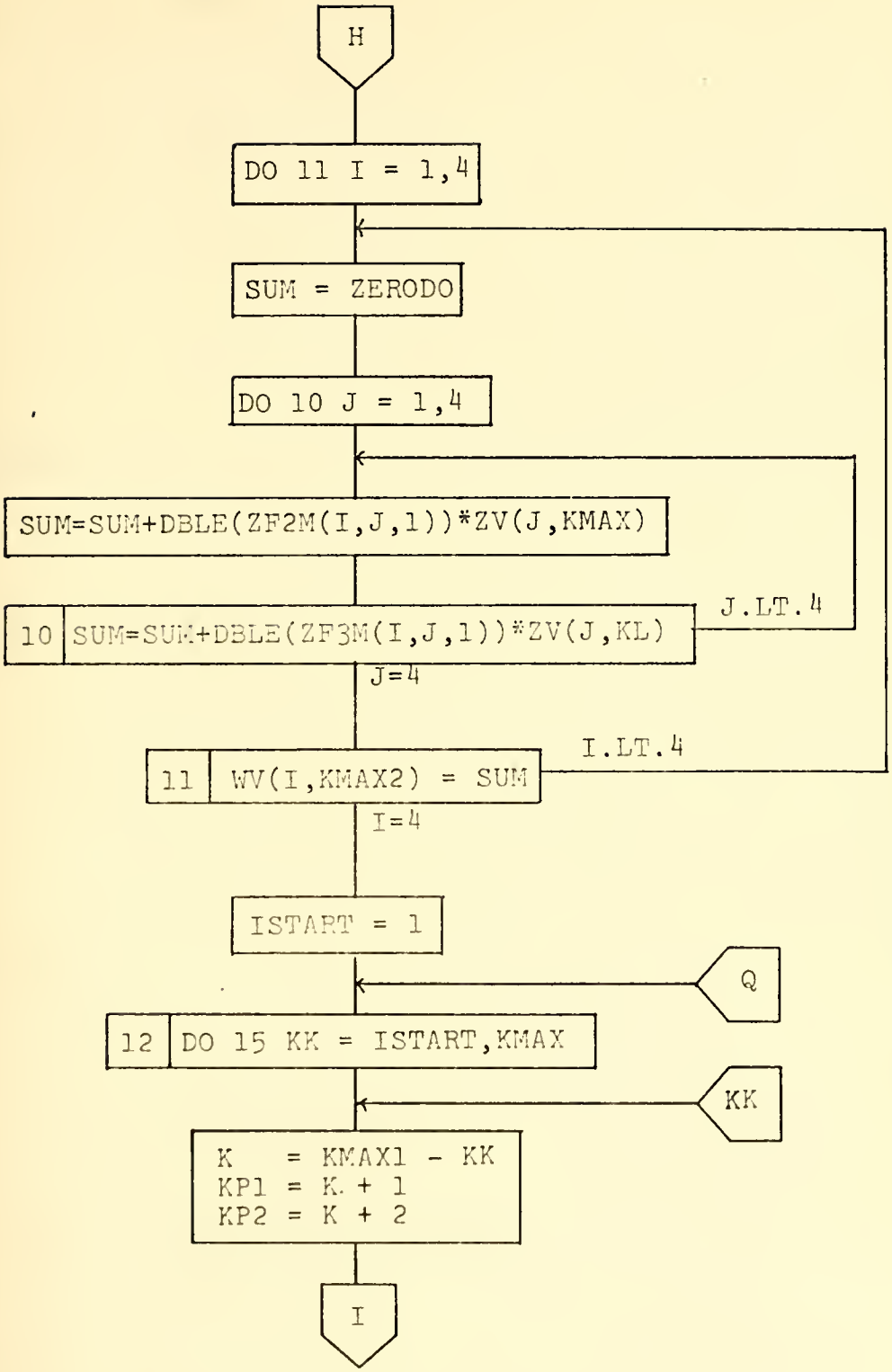


FIGURE 12 continued

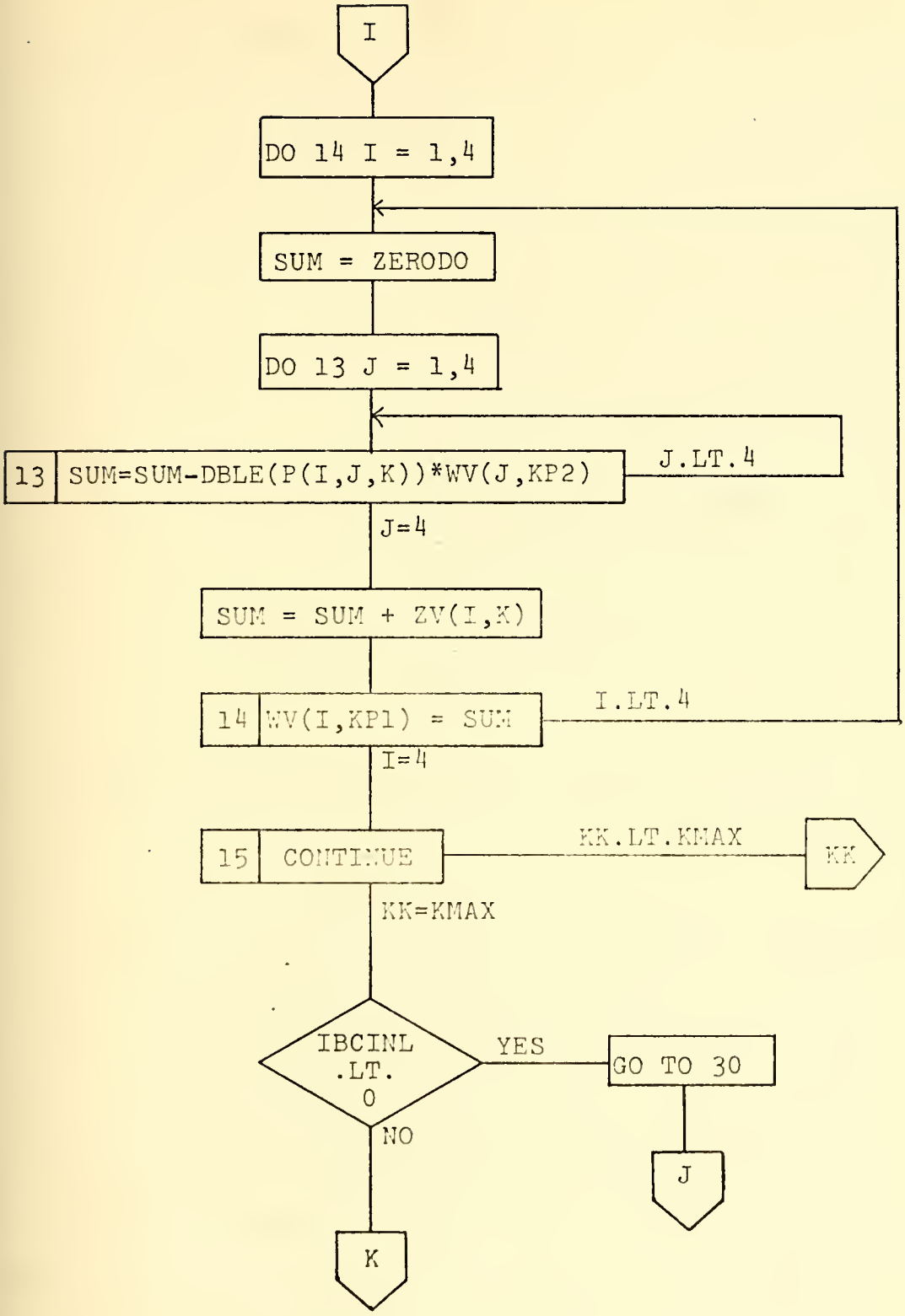


FIGURE 12 continued

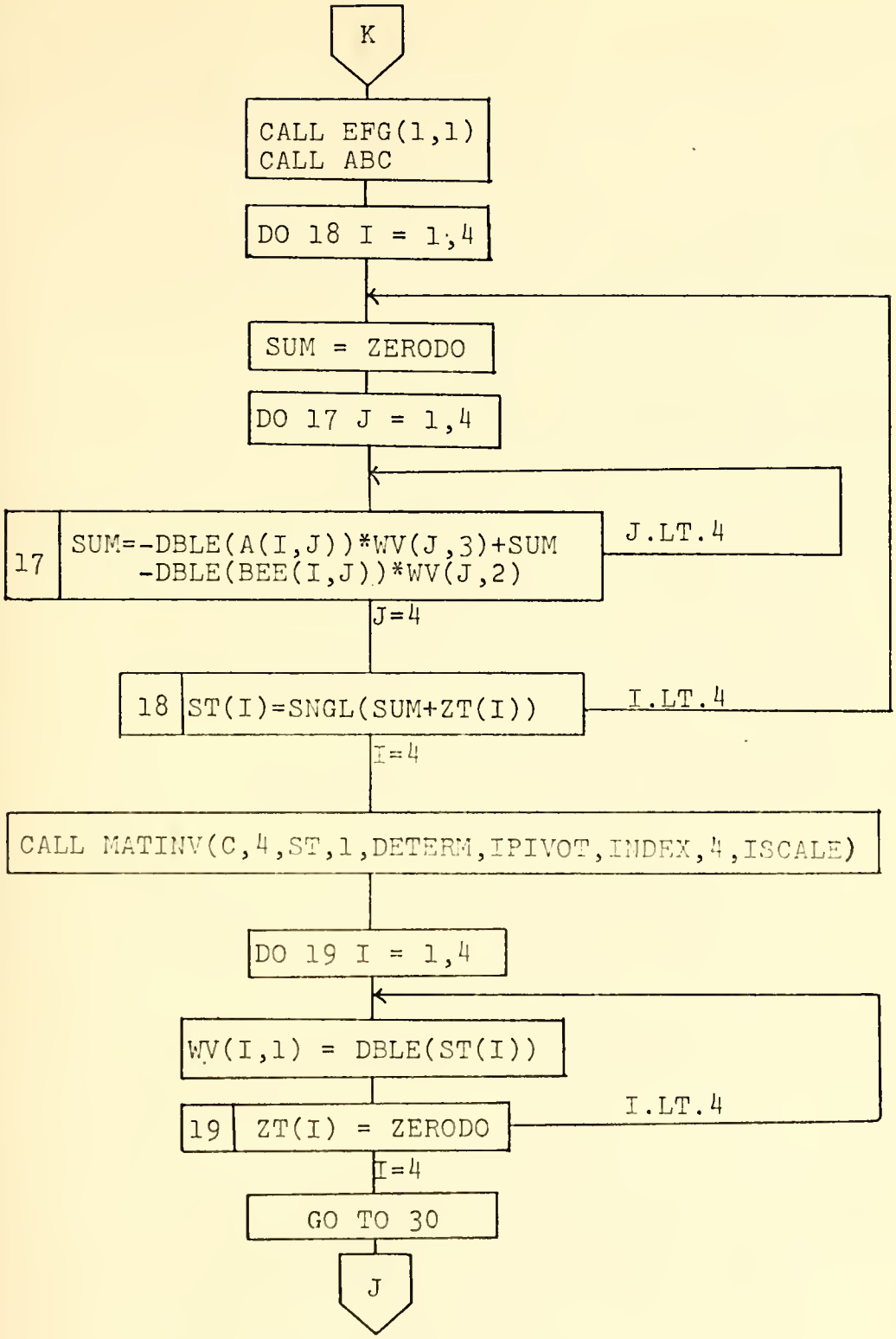


FIGURE 12 continued

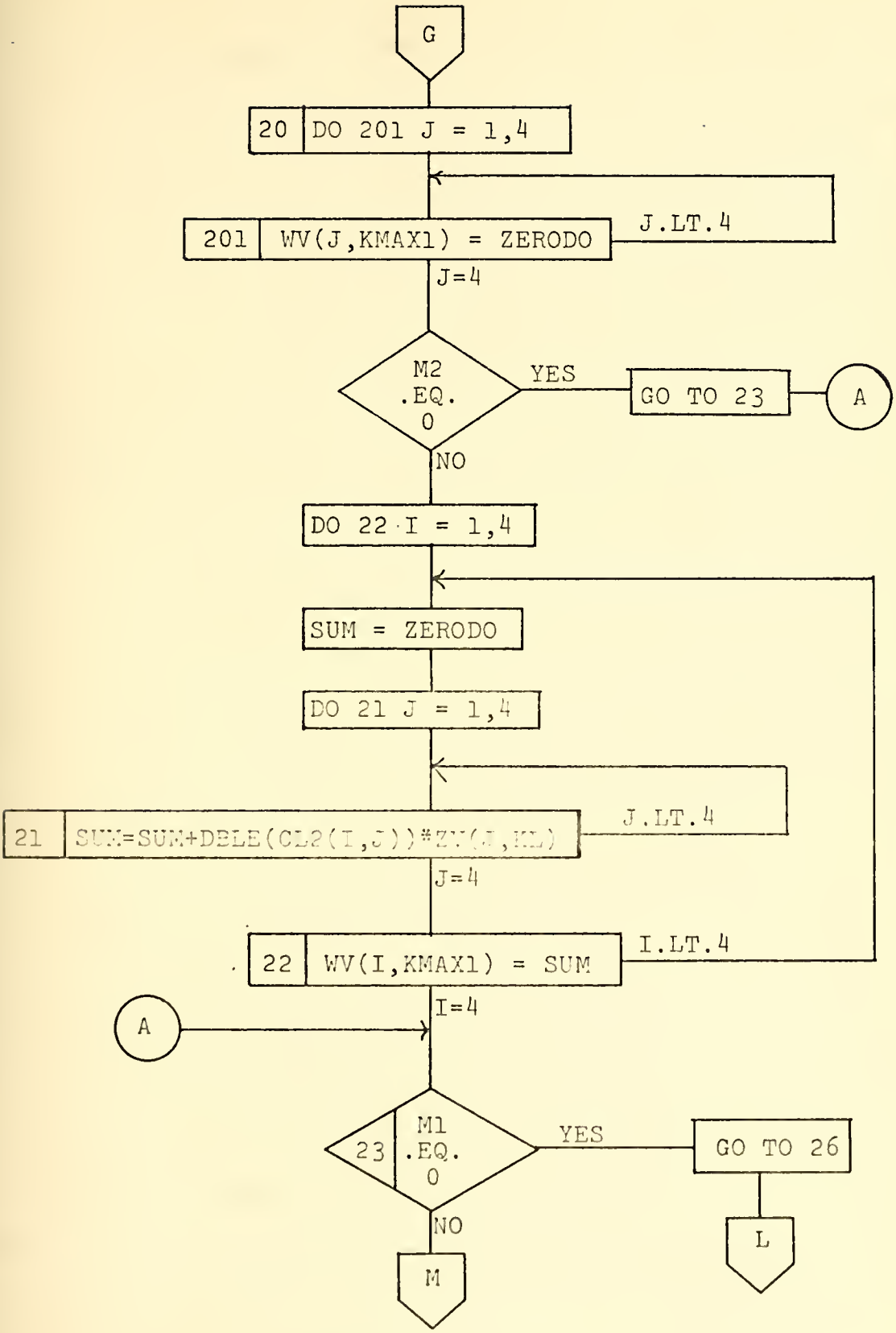


FIGURE 12 continued

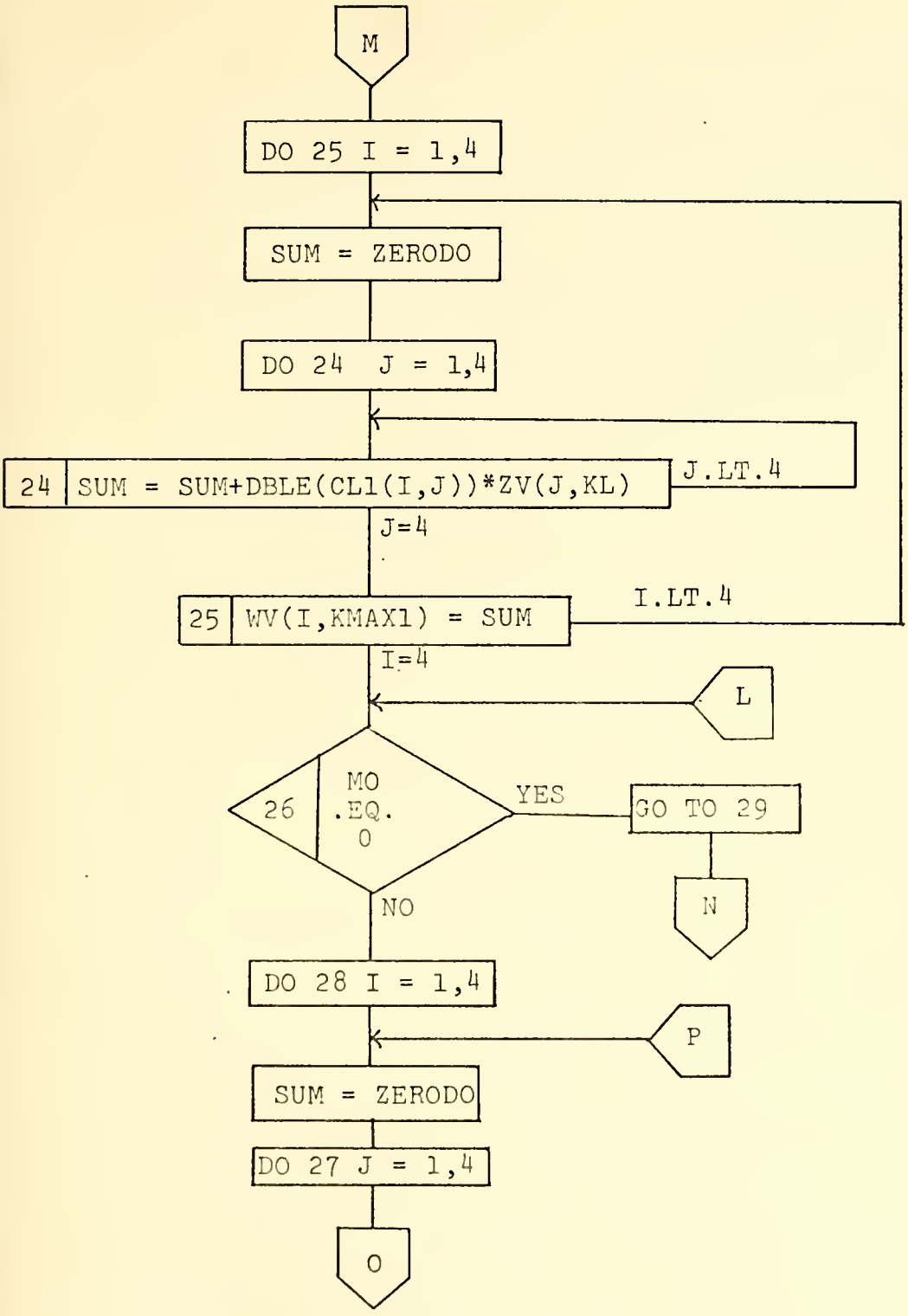


FIGURE 12 continued

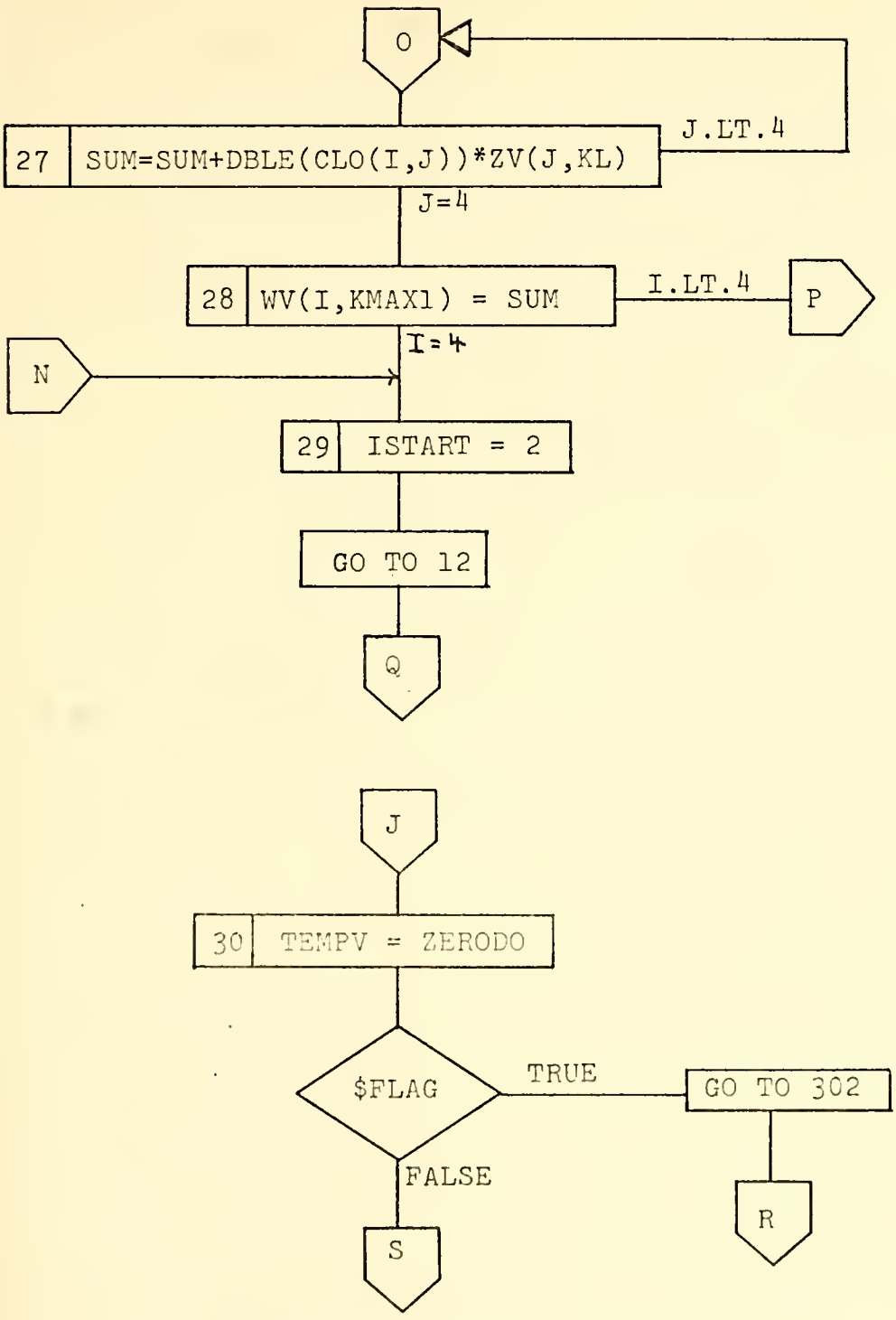


FIGURE 12 continued

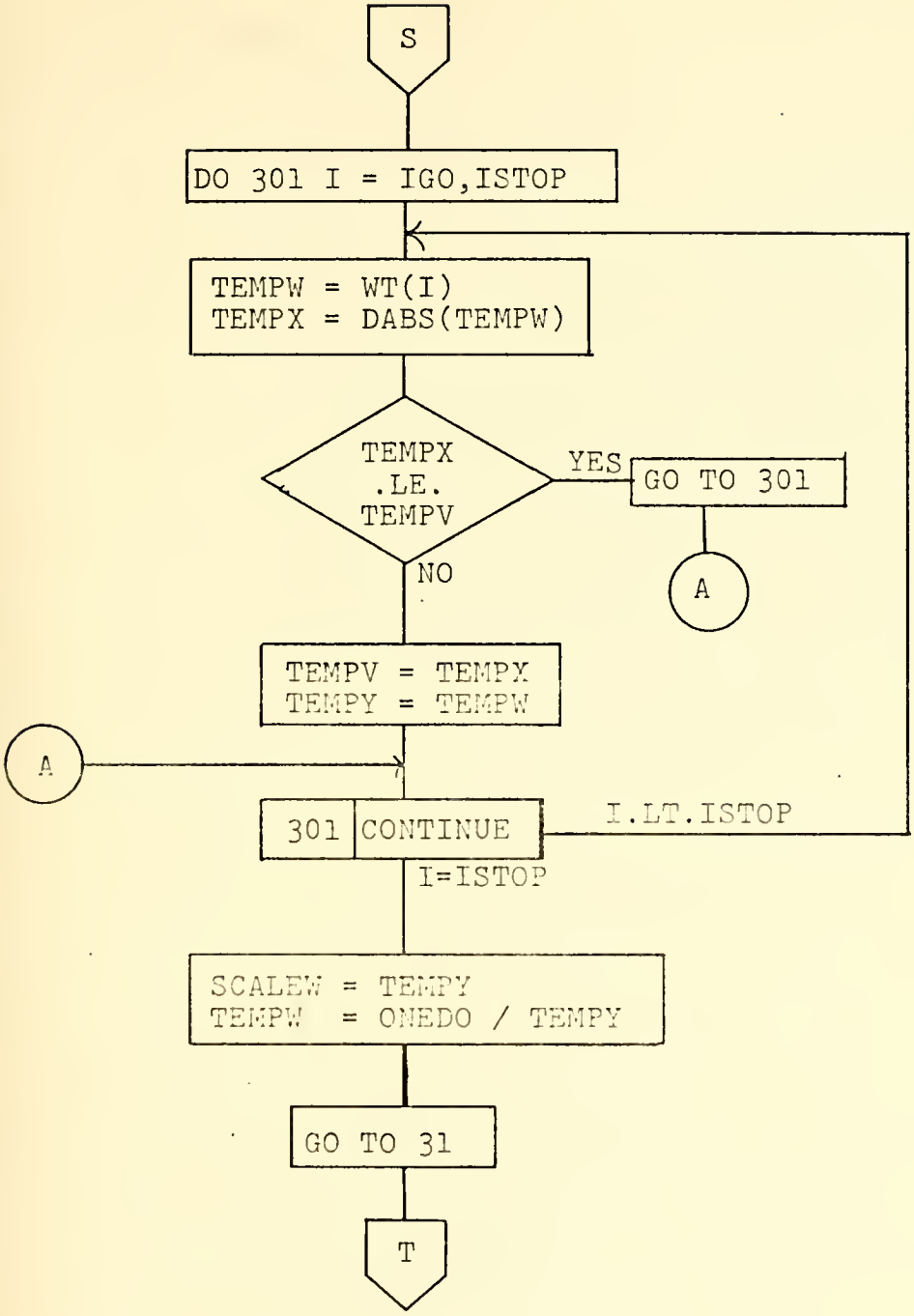


FIGURE 12 continued

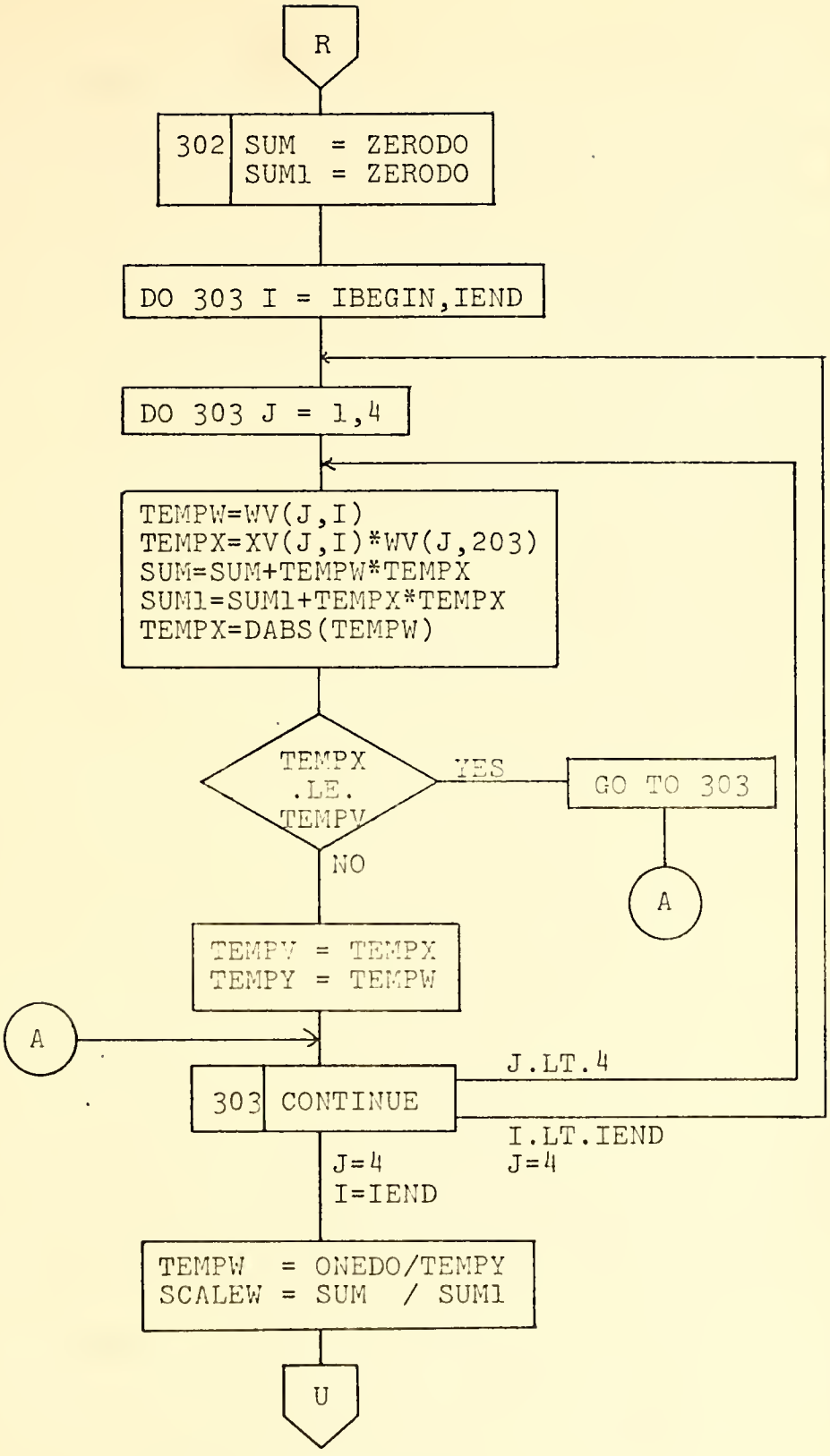


FIGURE 12 continued

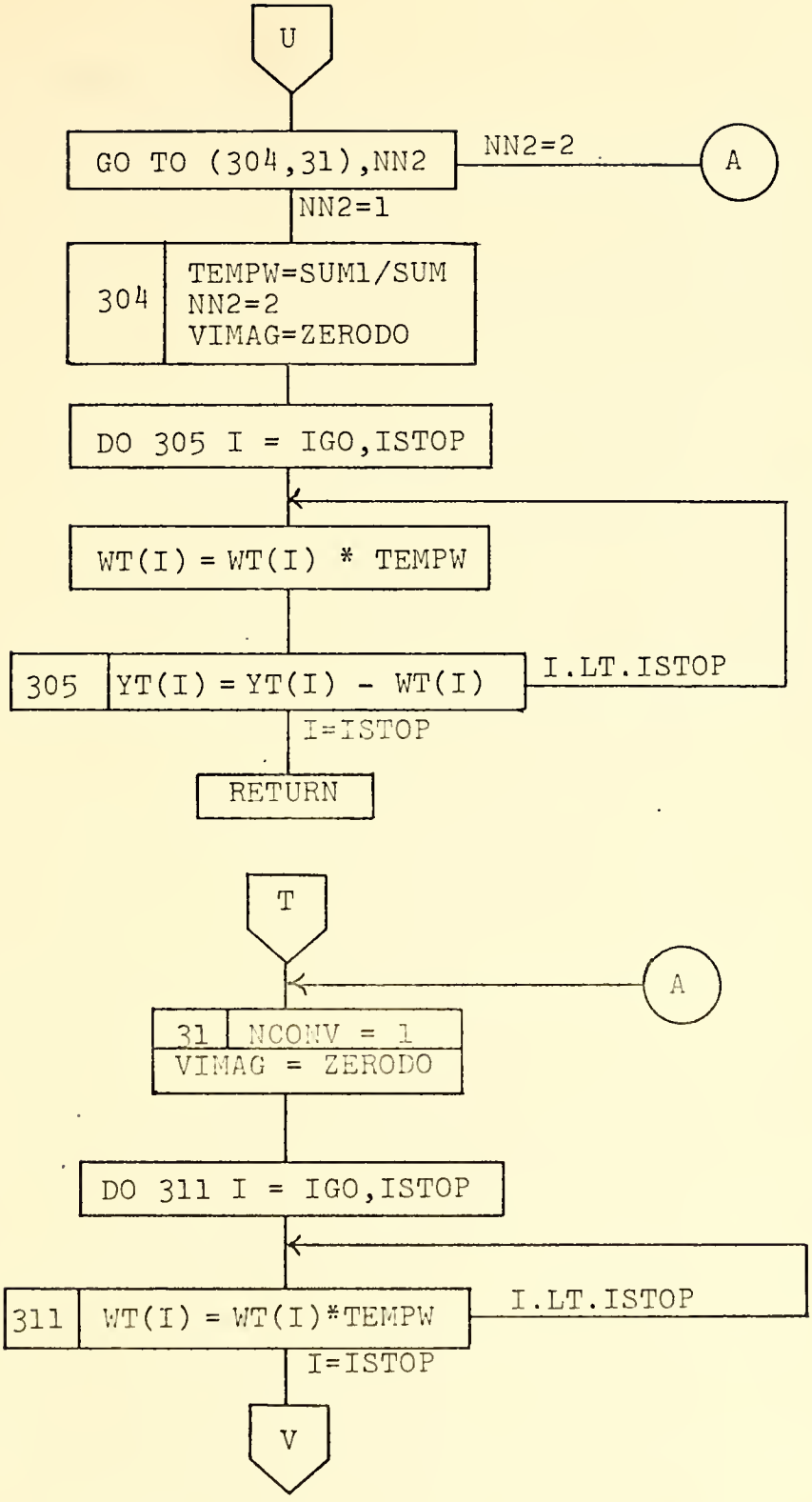


FIGURE 12 continued

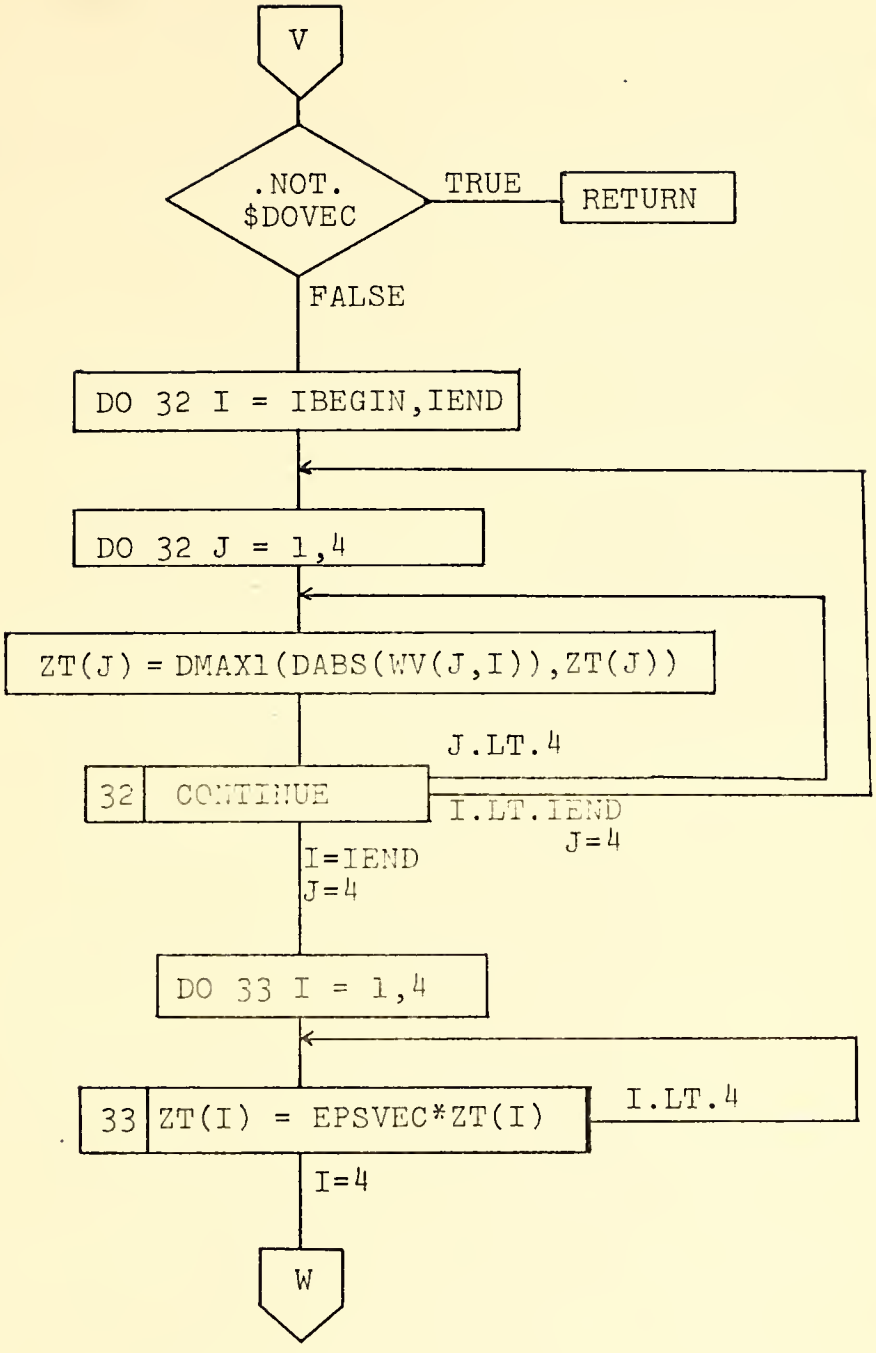


FIGURE 12 continued

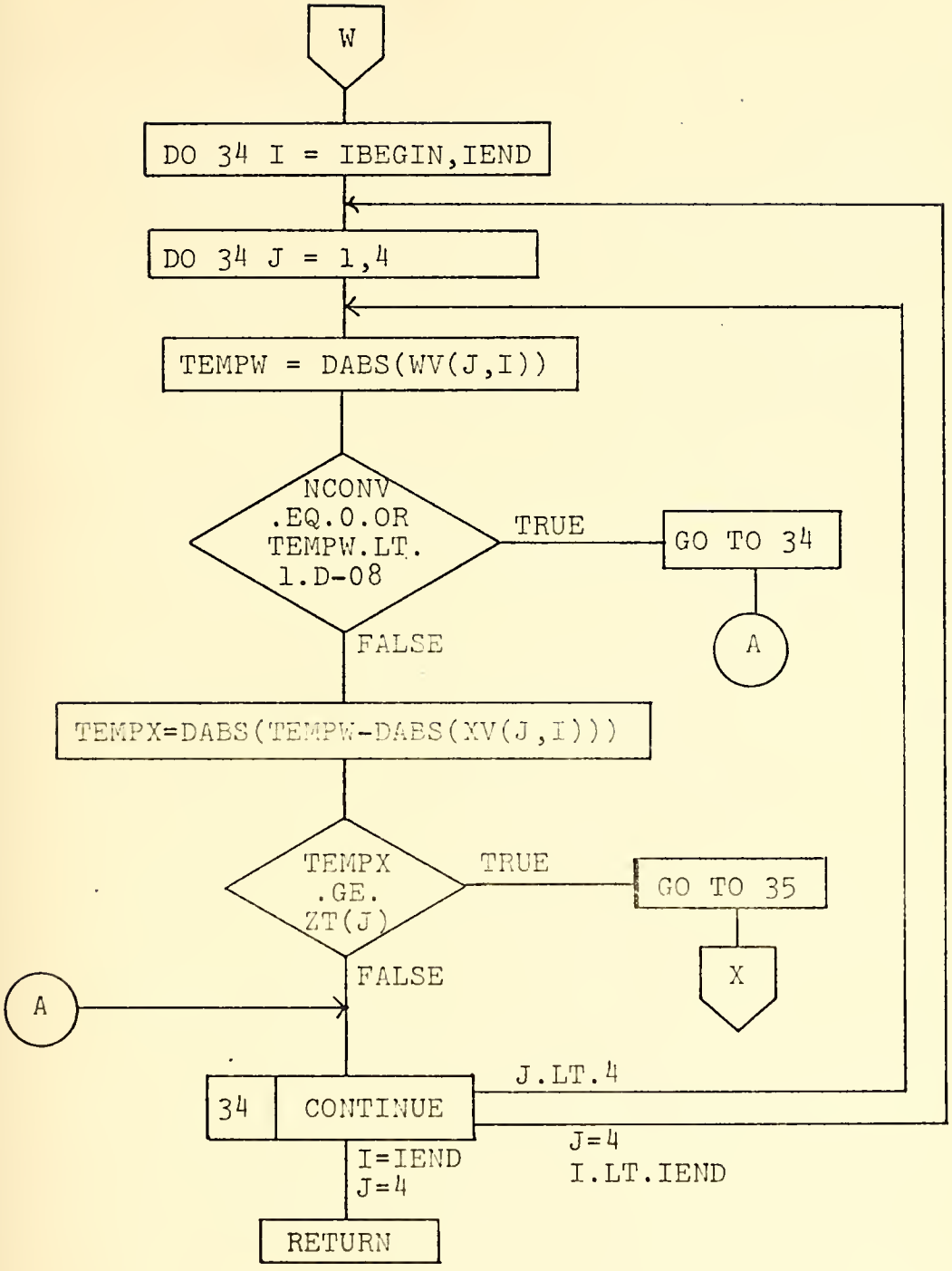


FIGURE 12 continued

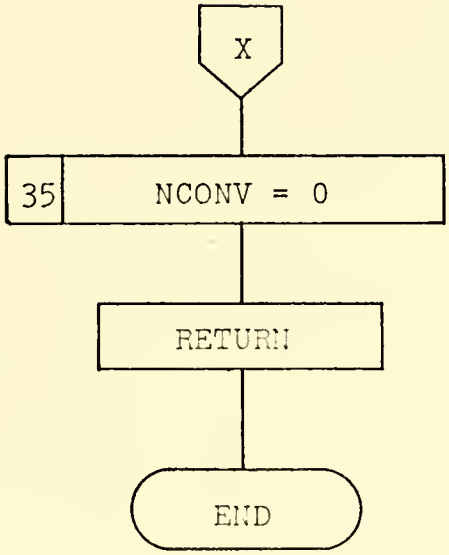


FIGURE 12 continued

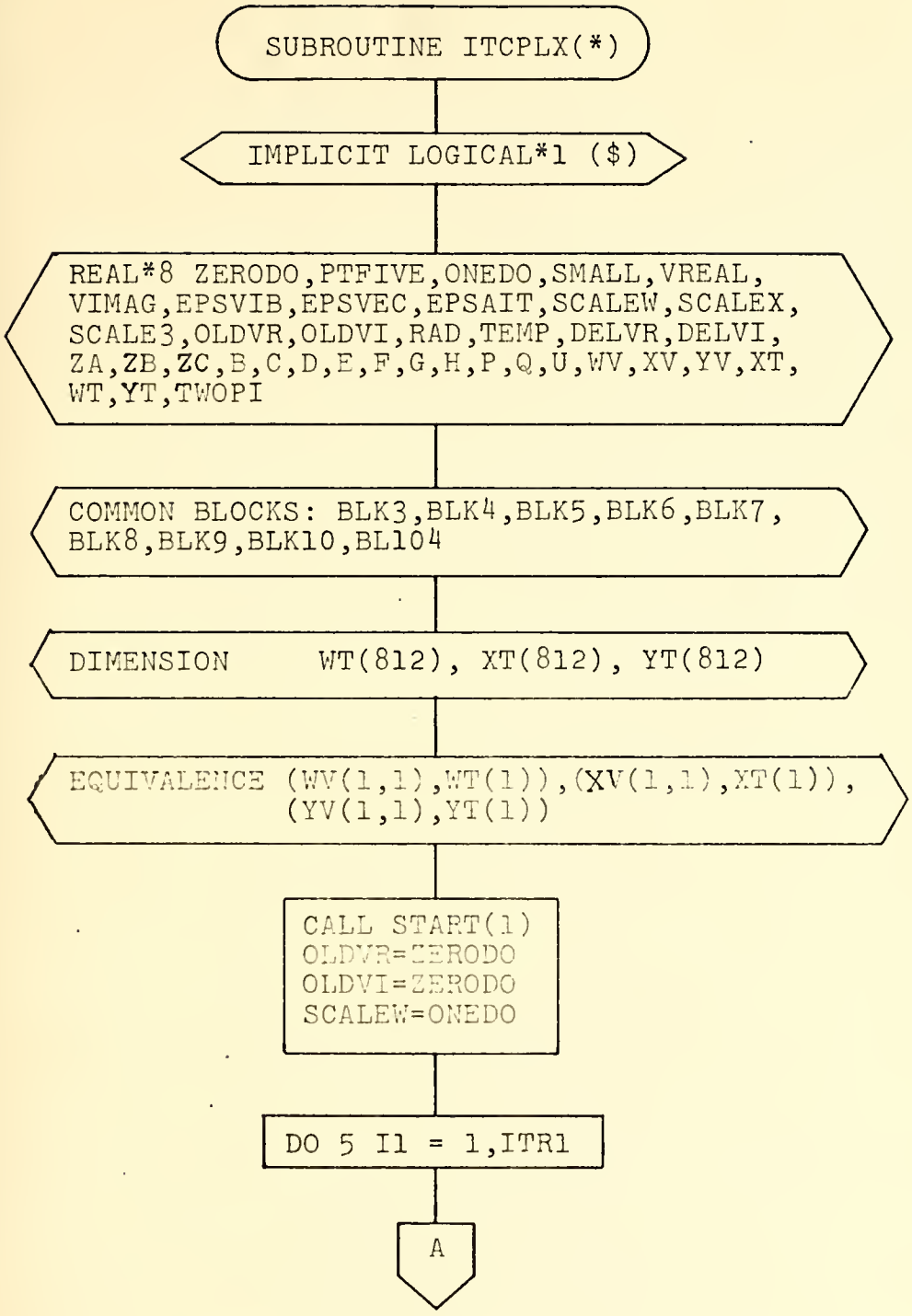


FIGURE 13 SUBROUTINE ITCPLX(*) FLOWCHART

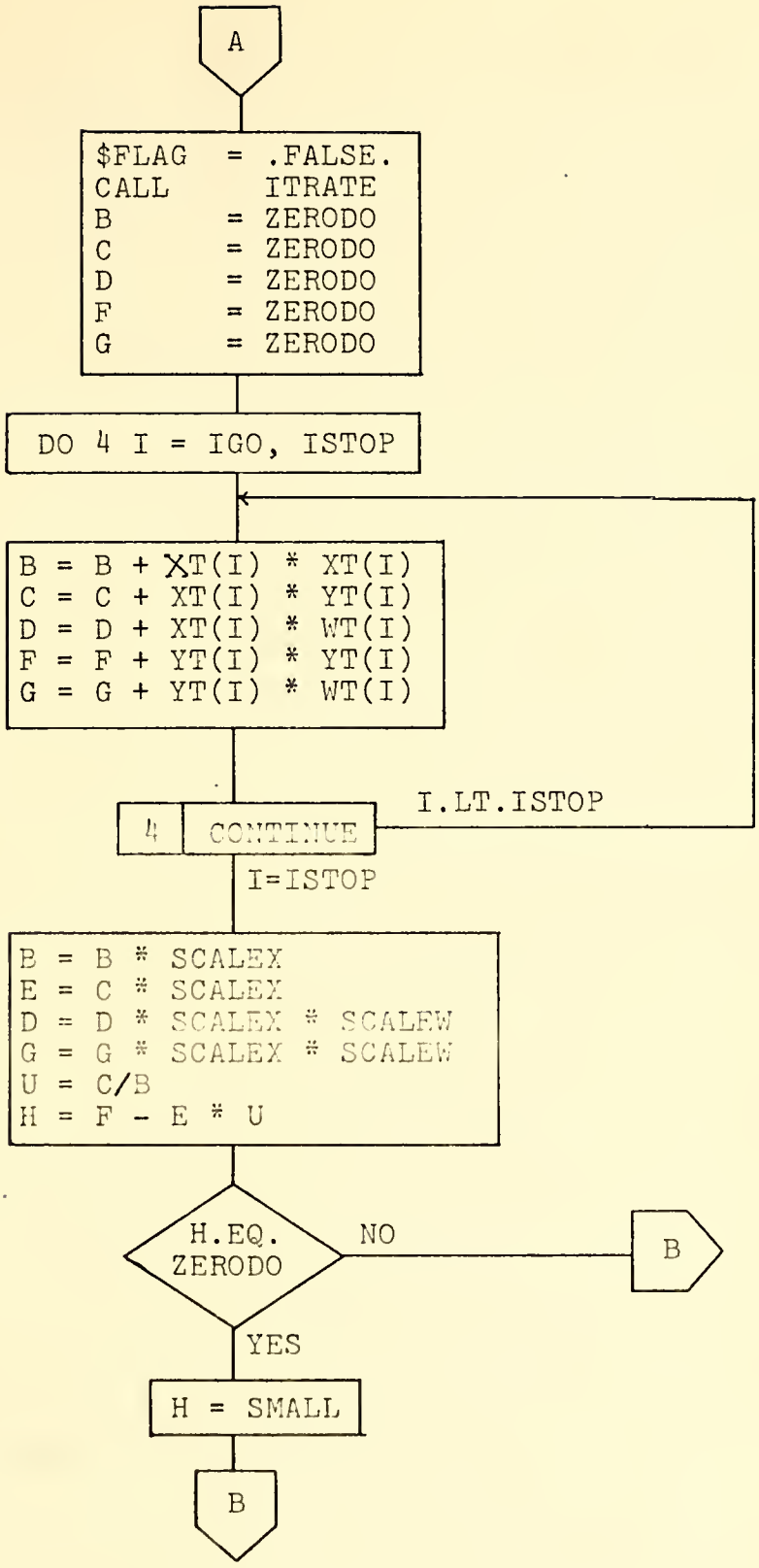


FIGURE 13 continued

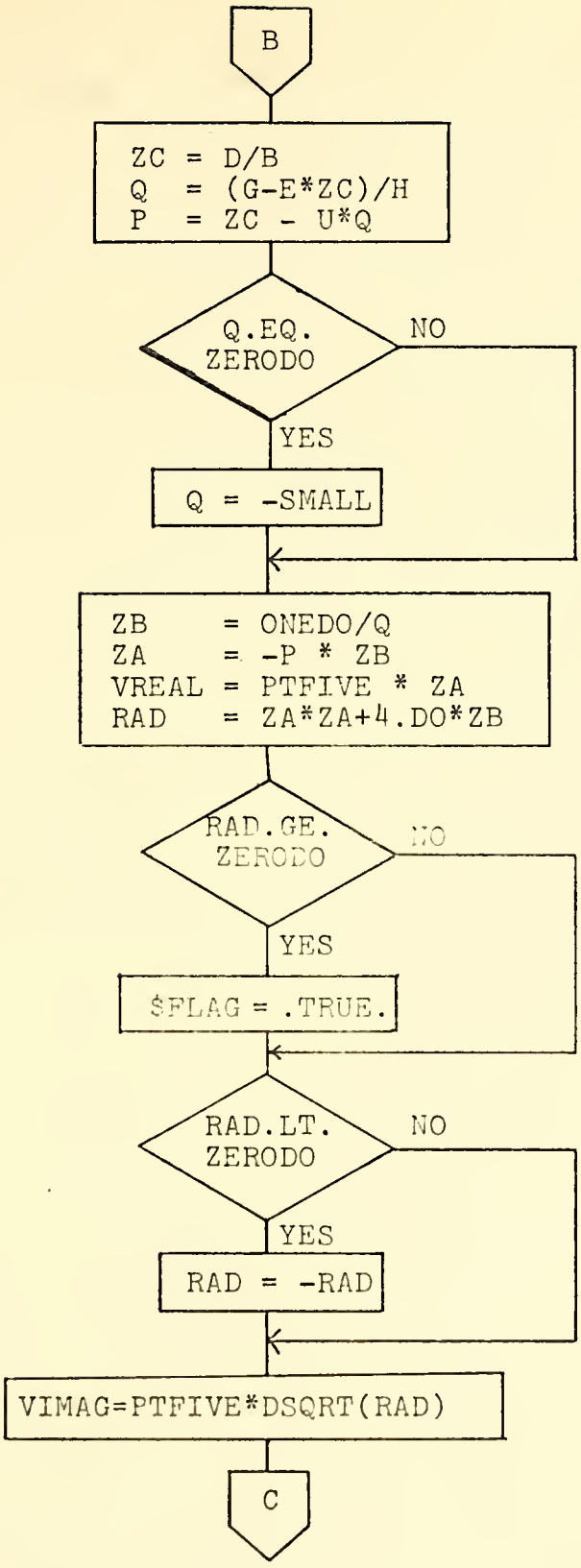


FIGURE 13 continued

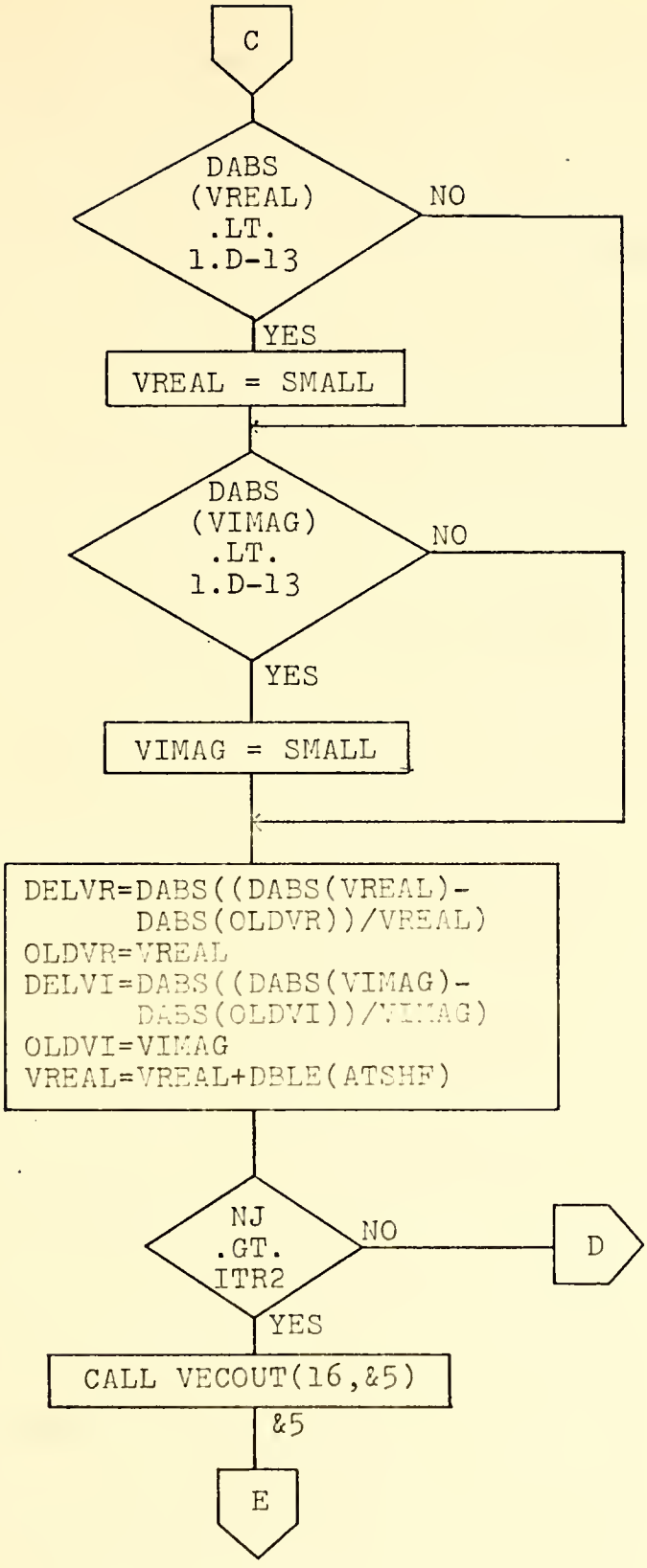


FIGURE 13 continued

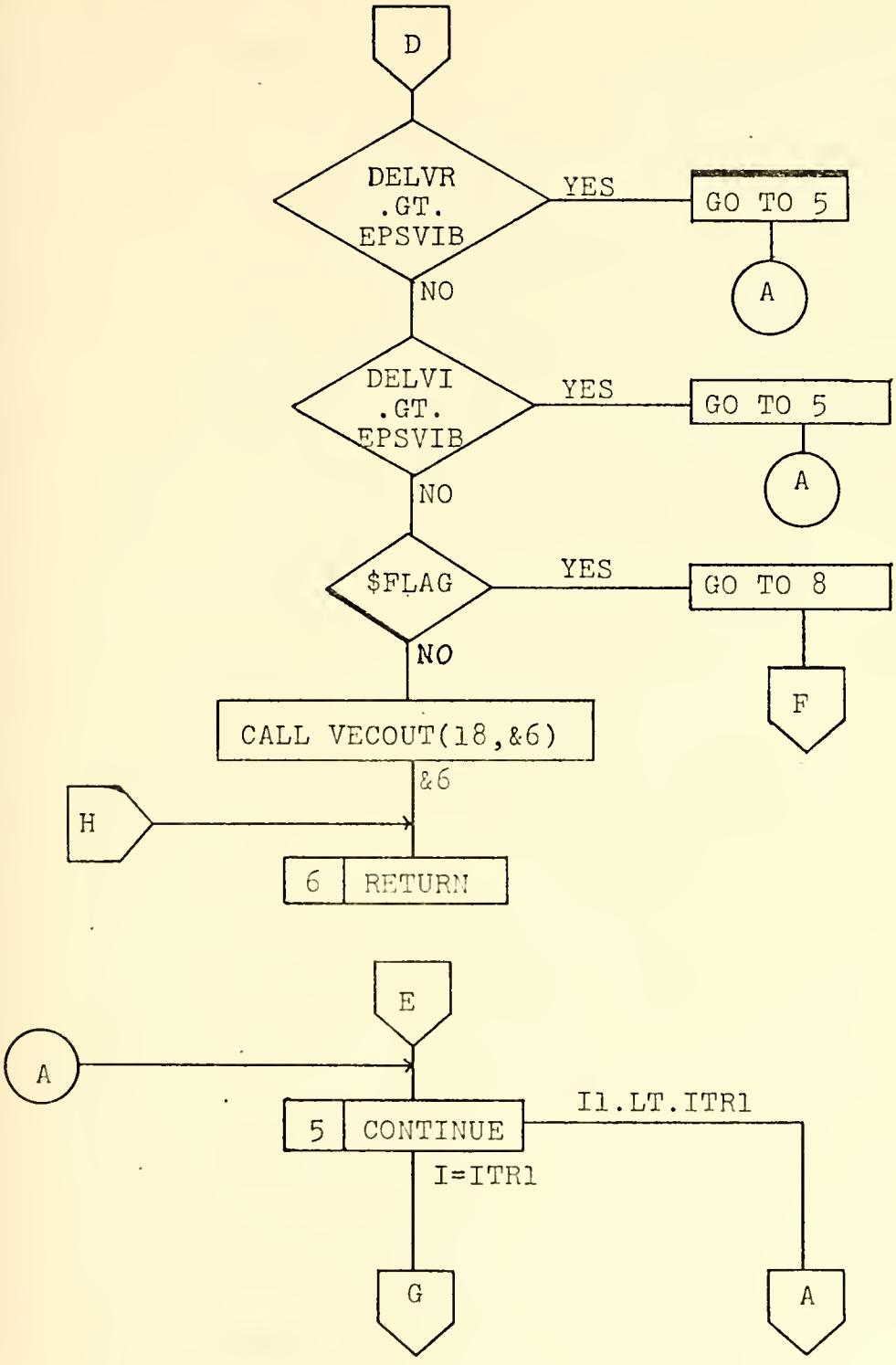


FIGURE 13 continued

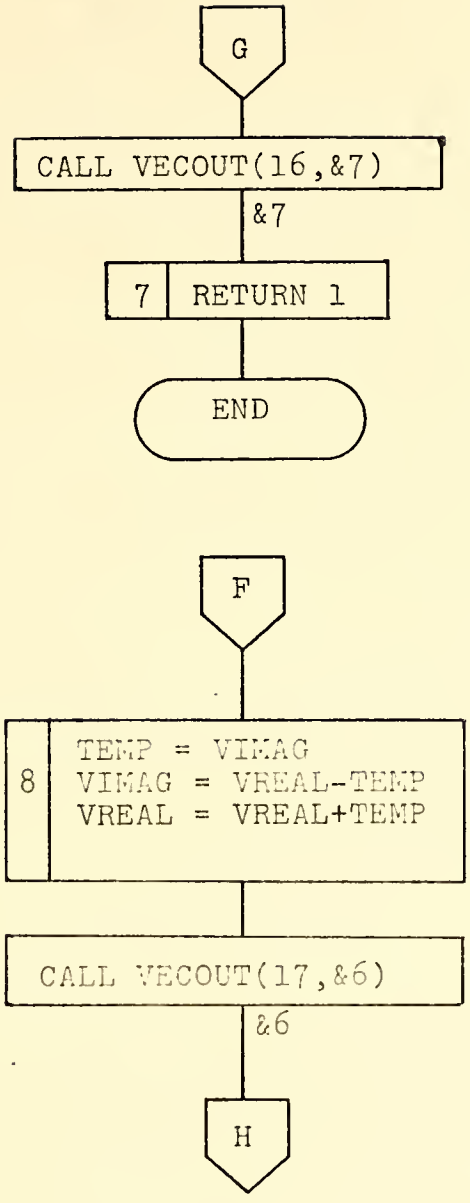


FIGURE 13 continued

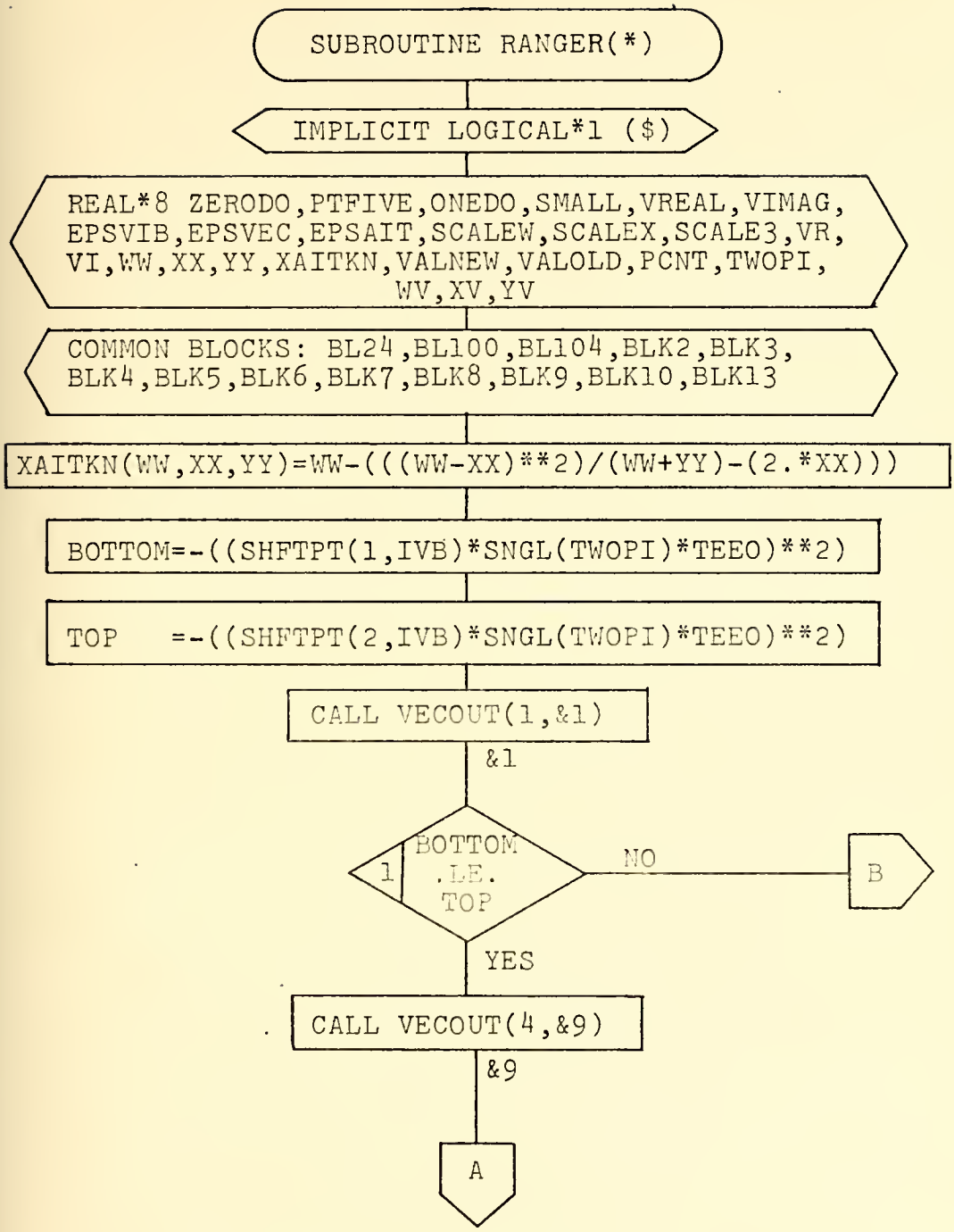


FIGURE 14. SUBROUTINE RANGER(*) FLOWCHART

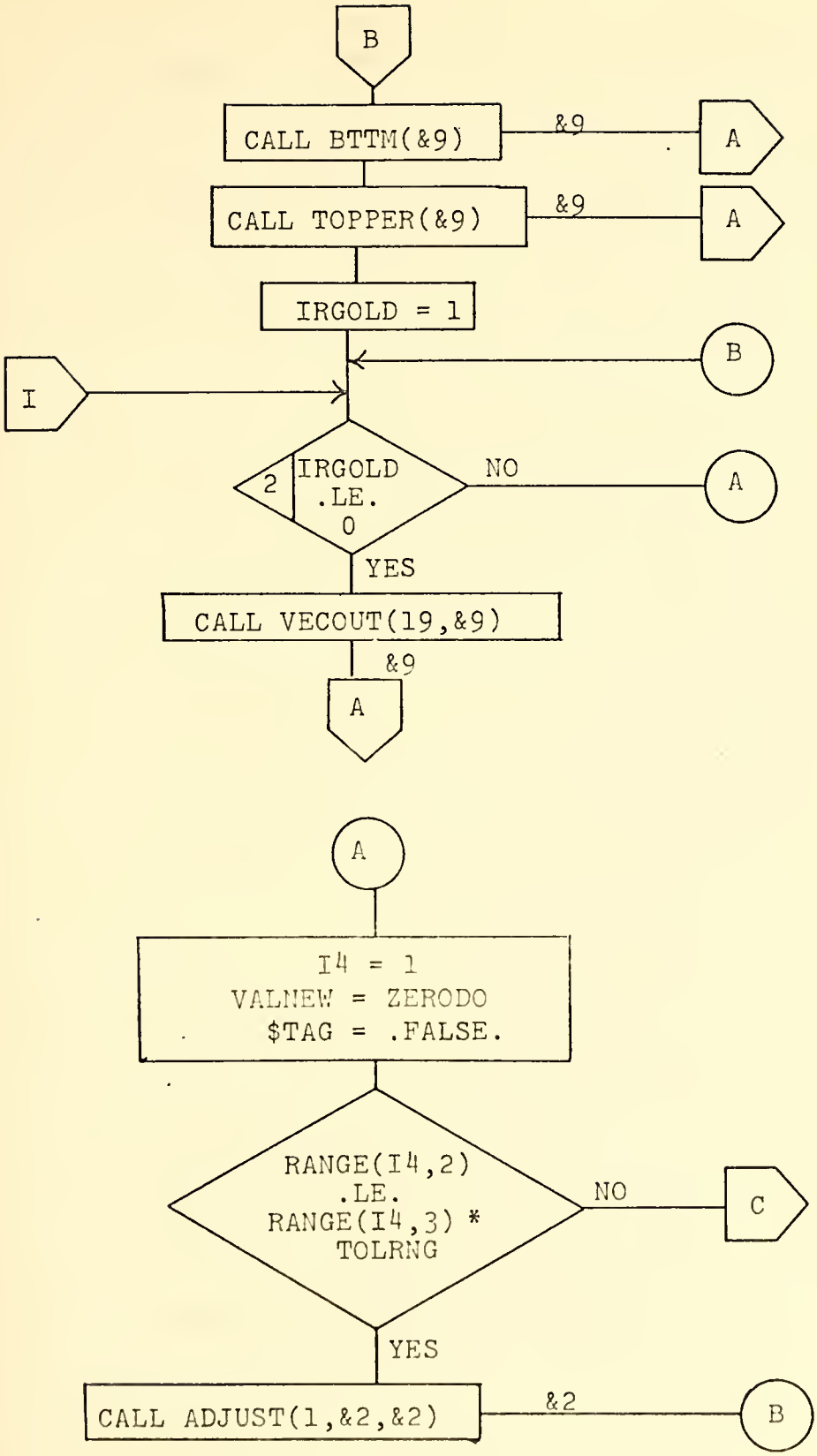


FIGURE 14 continued

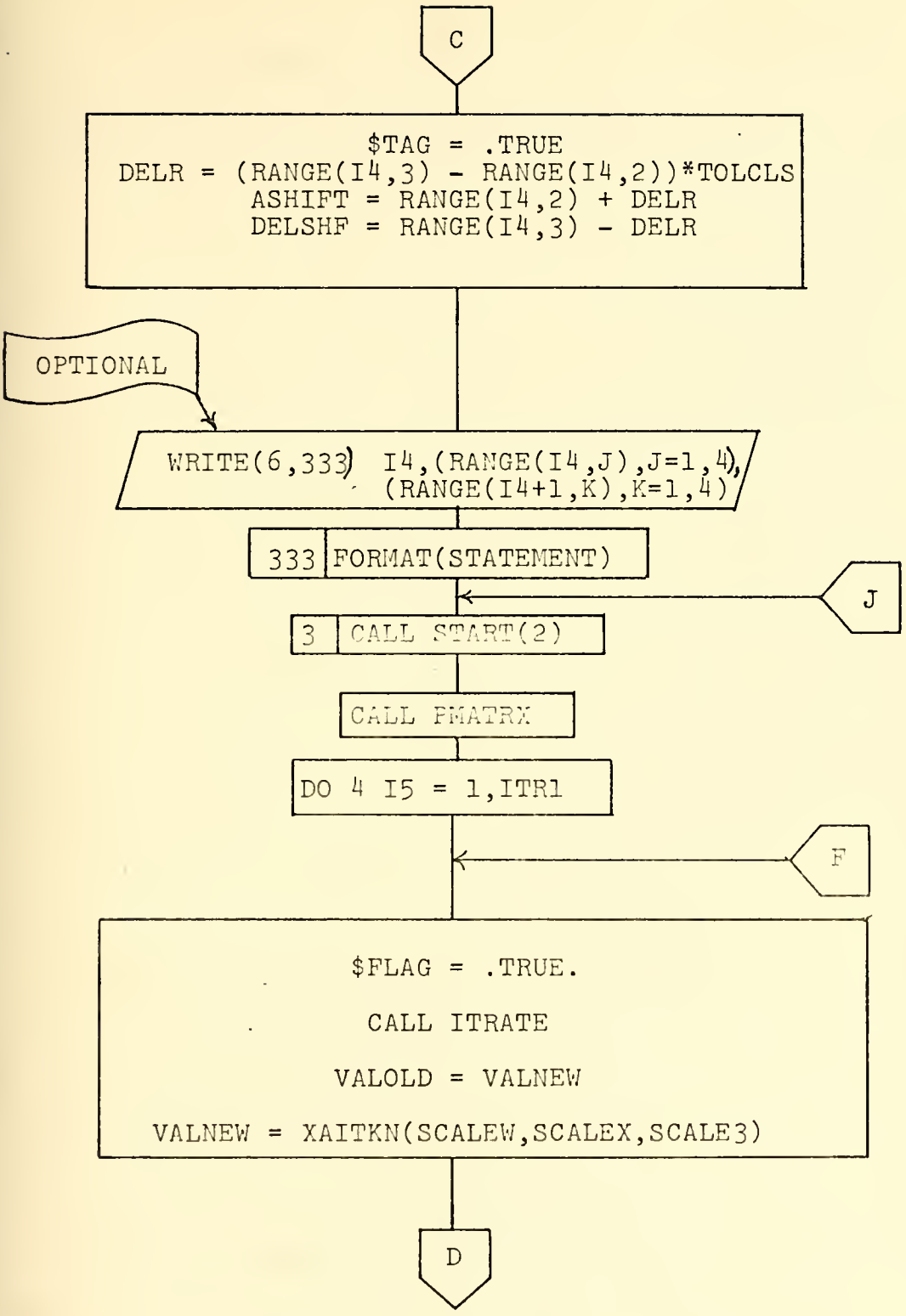


FIGURE 14 continued

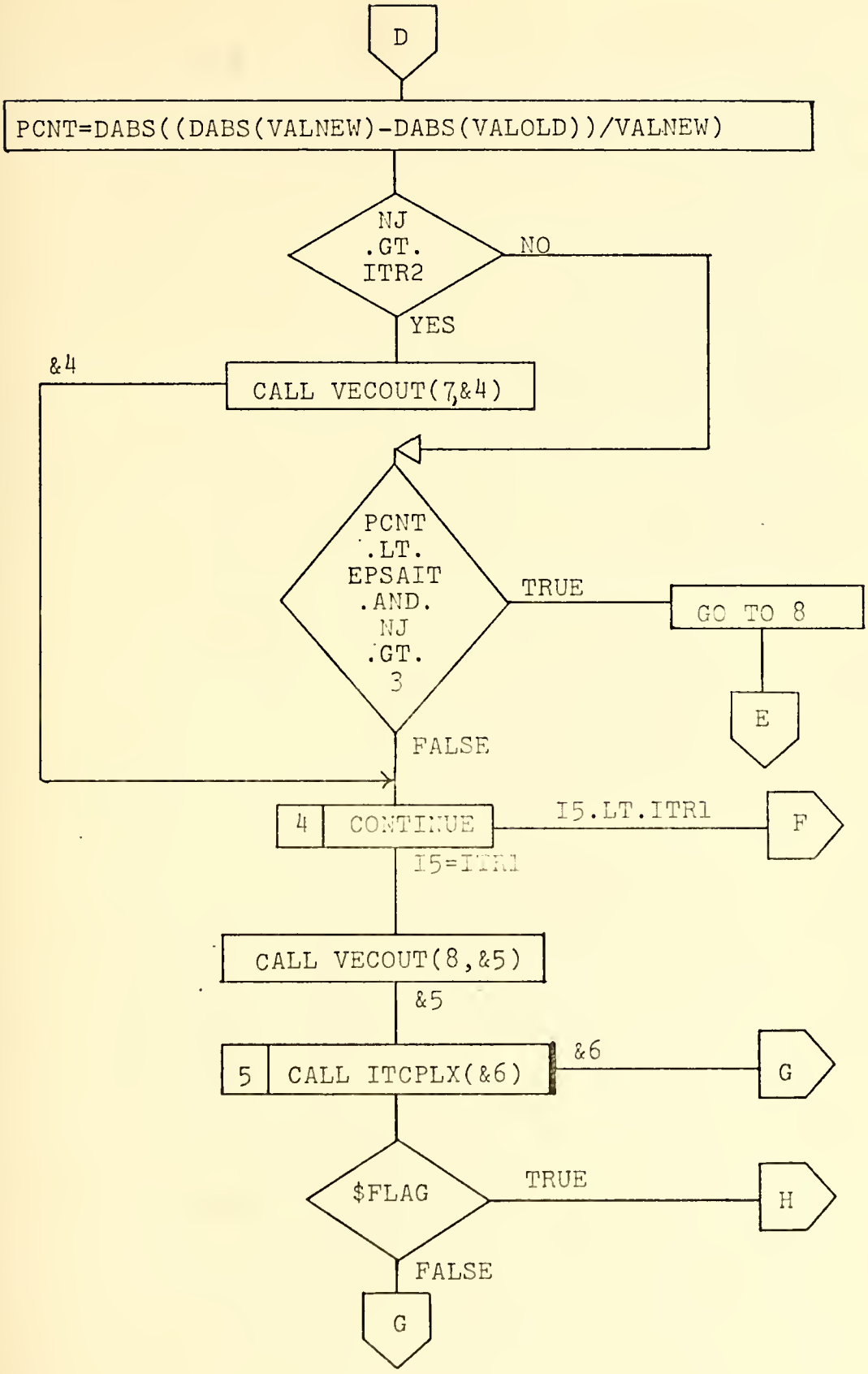


FIGURE 14 continued

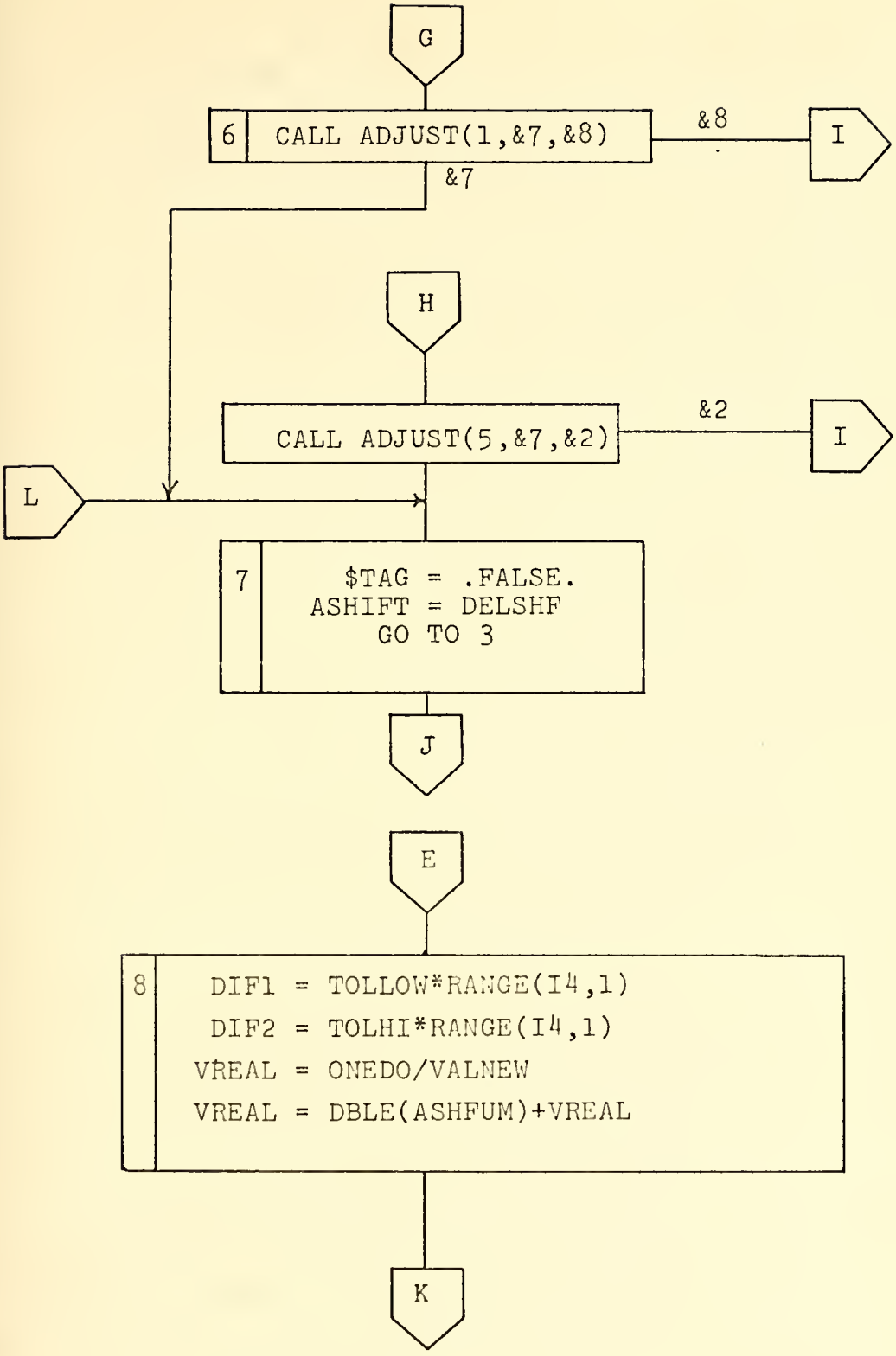


FIGURE 14 continued

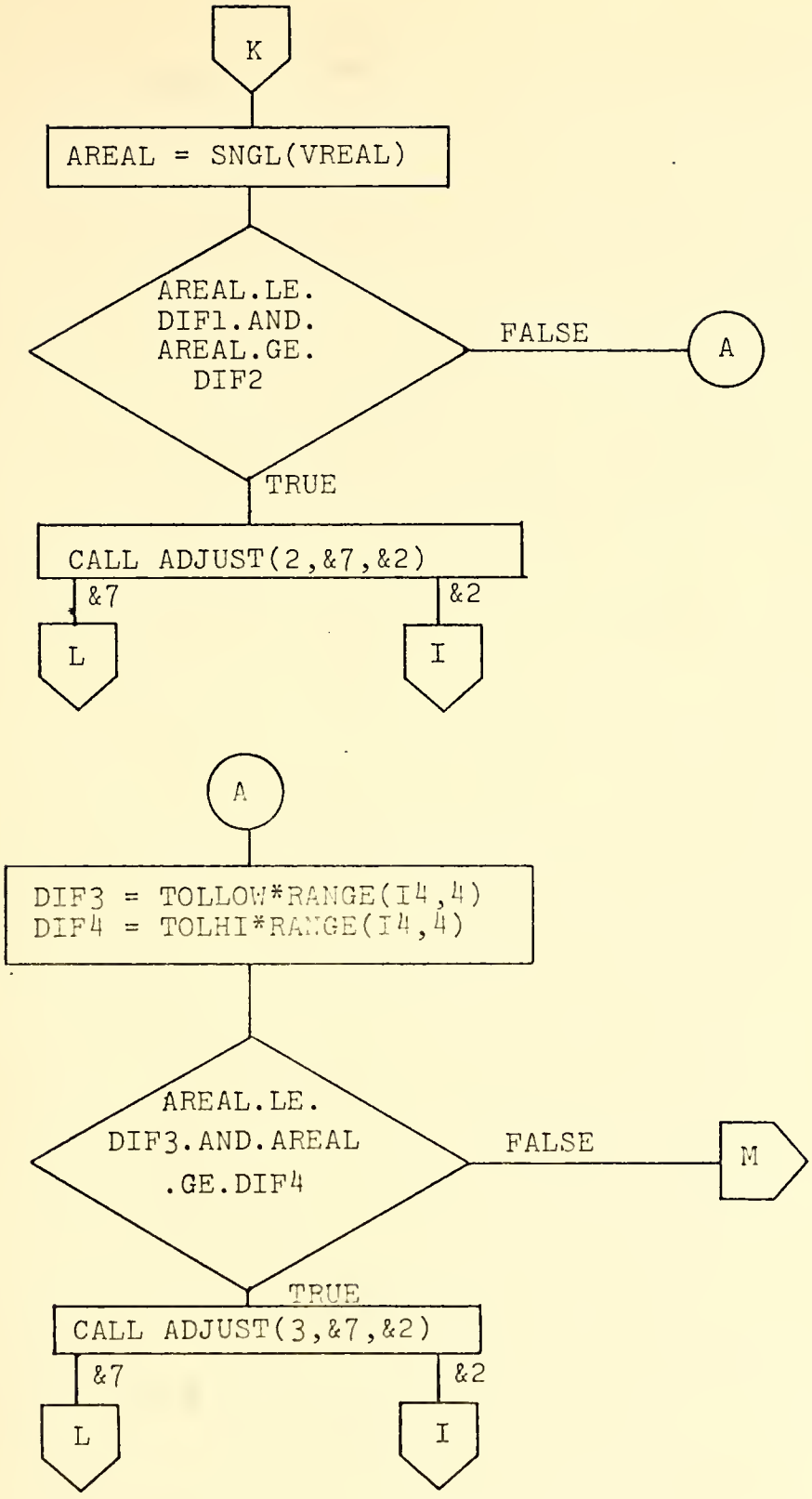


FIGURE 14 continued

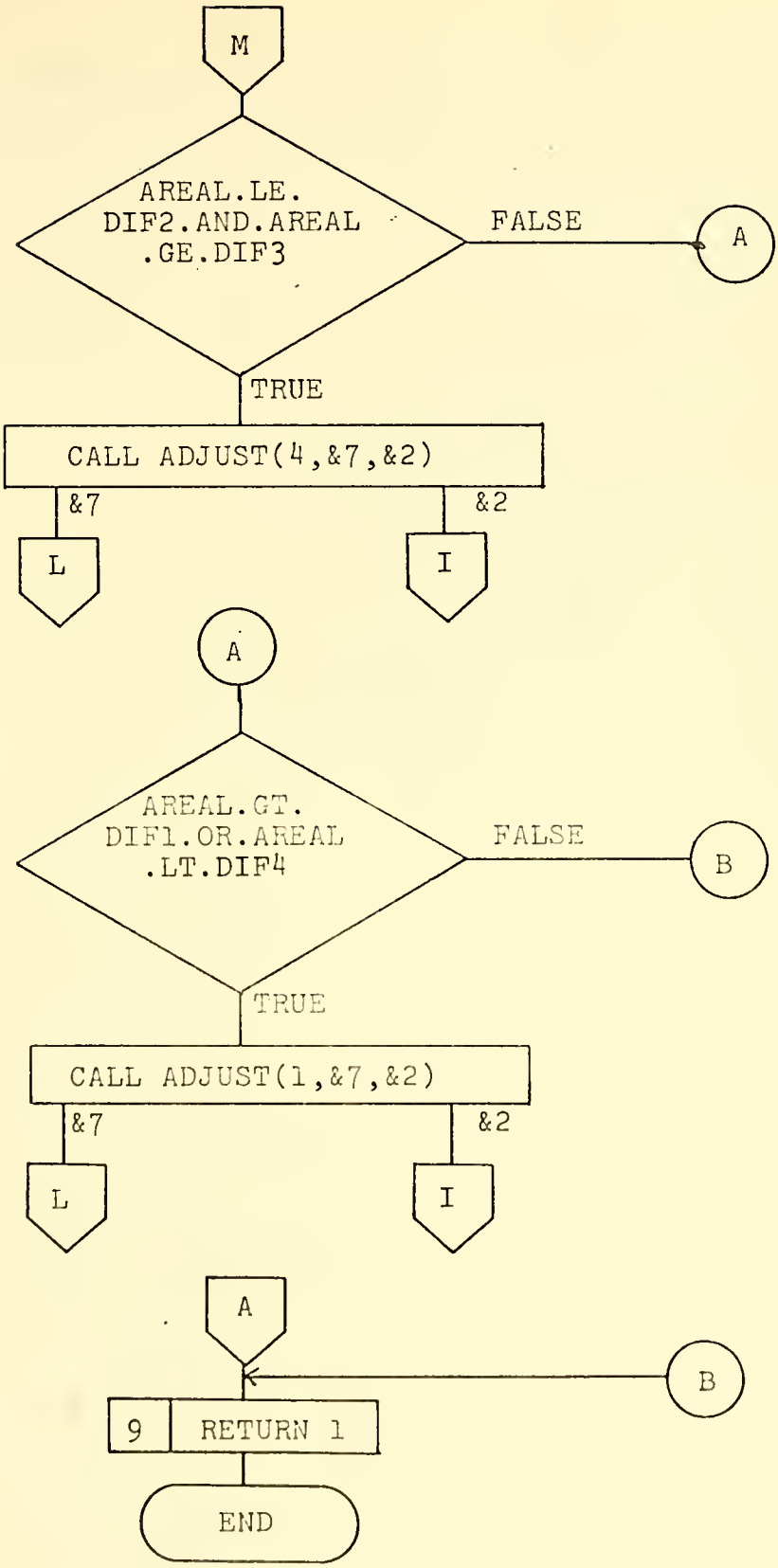


FIGURE 14 continued

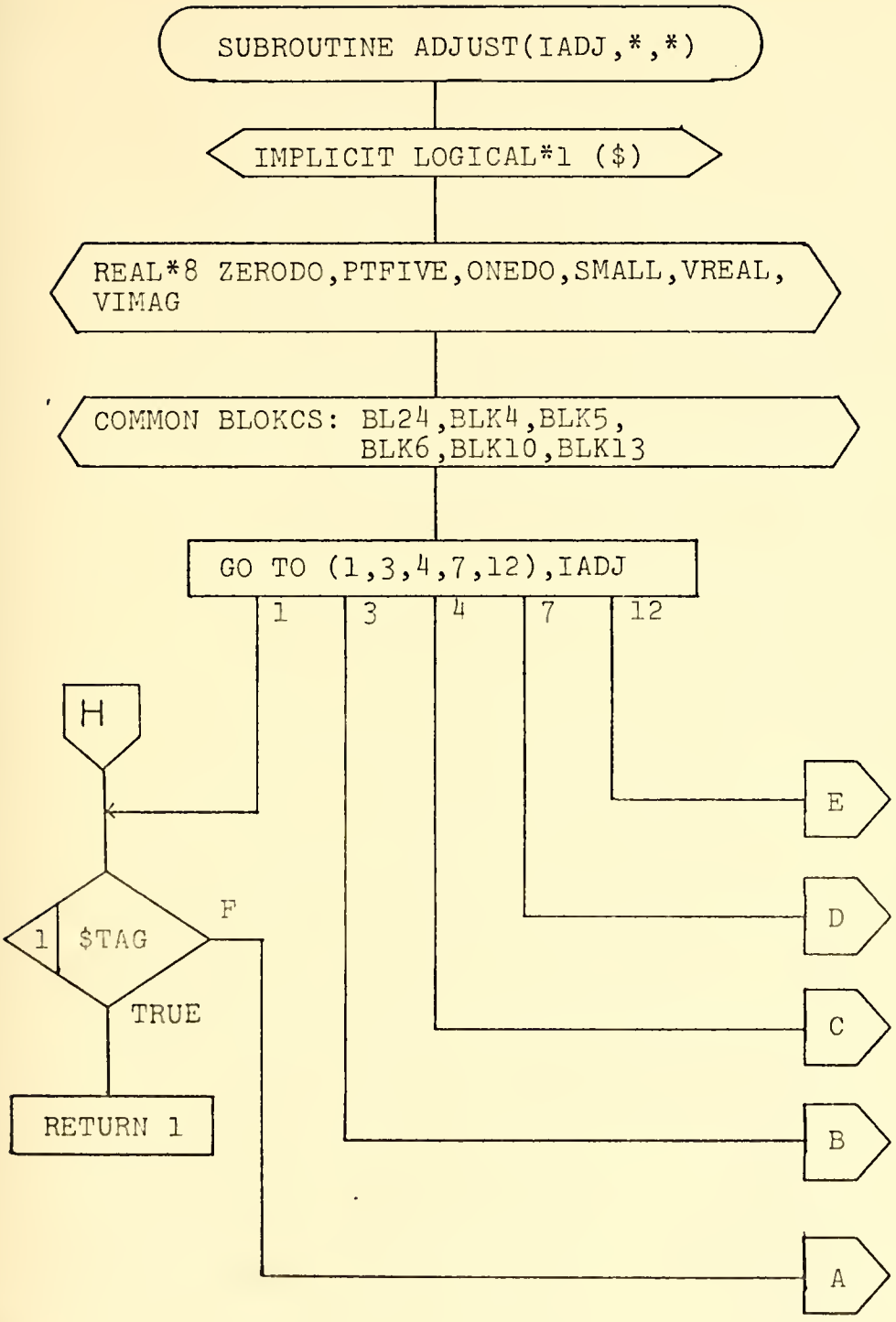


FIGURE 15. SUBROUTINE ADJUST(IADJ,*,*) FLOWCHART

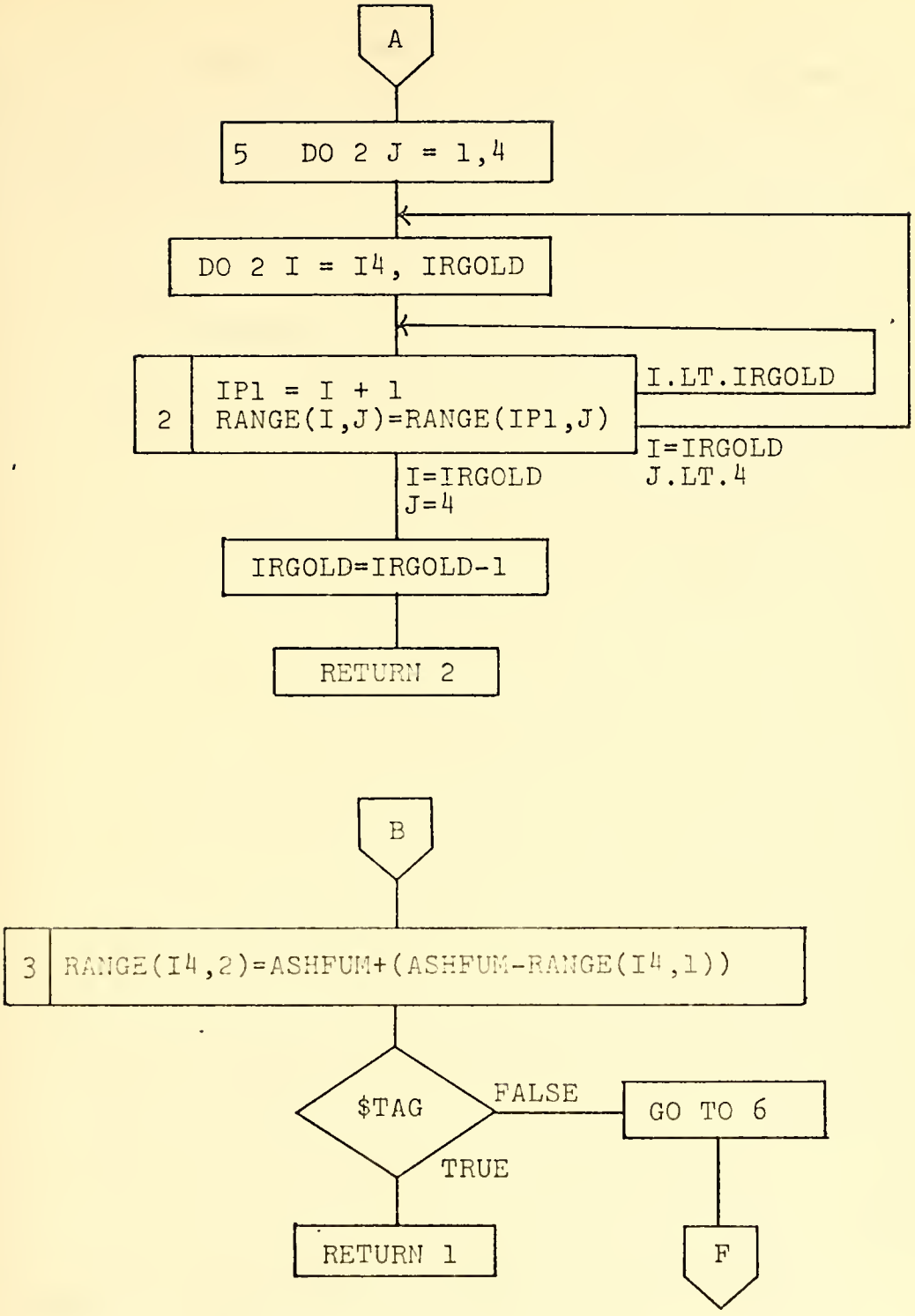
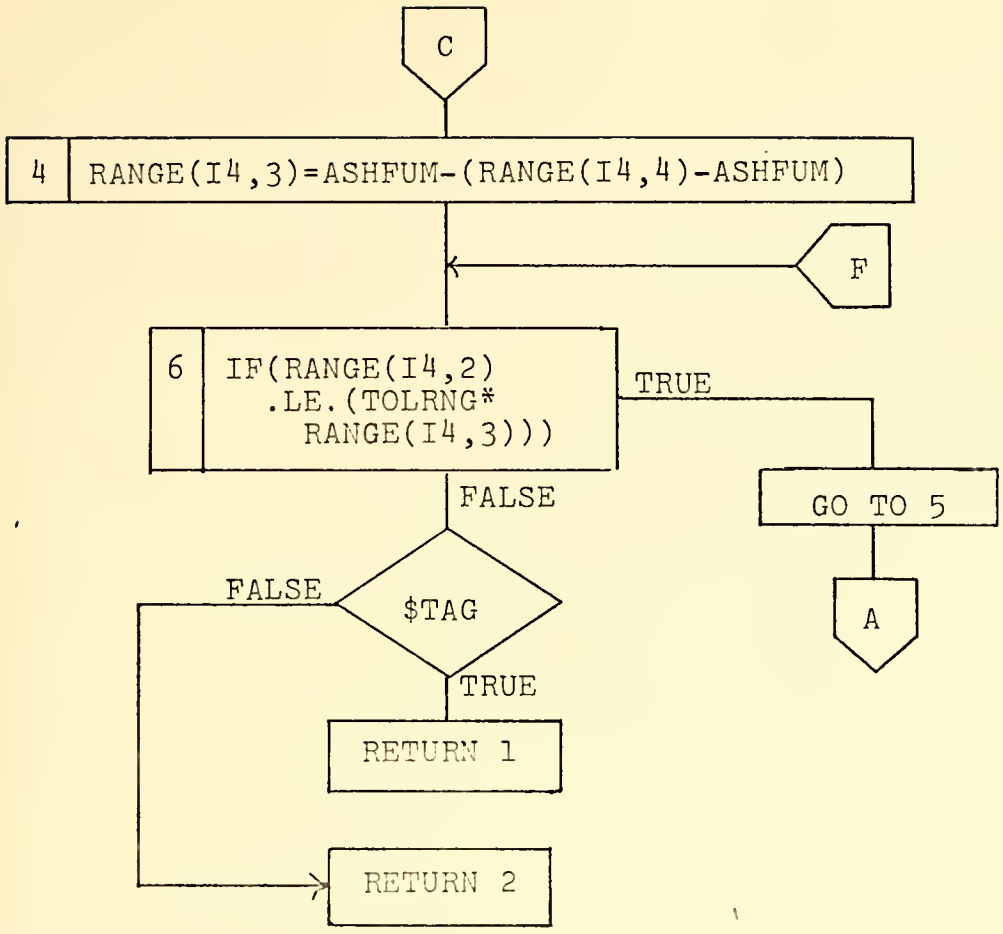


FIGURE 15 continued



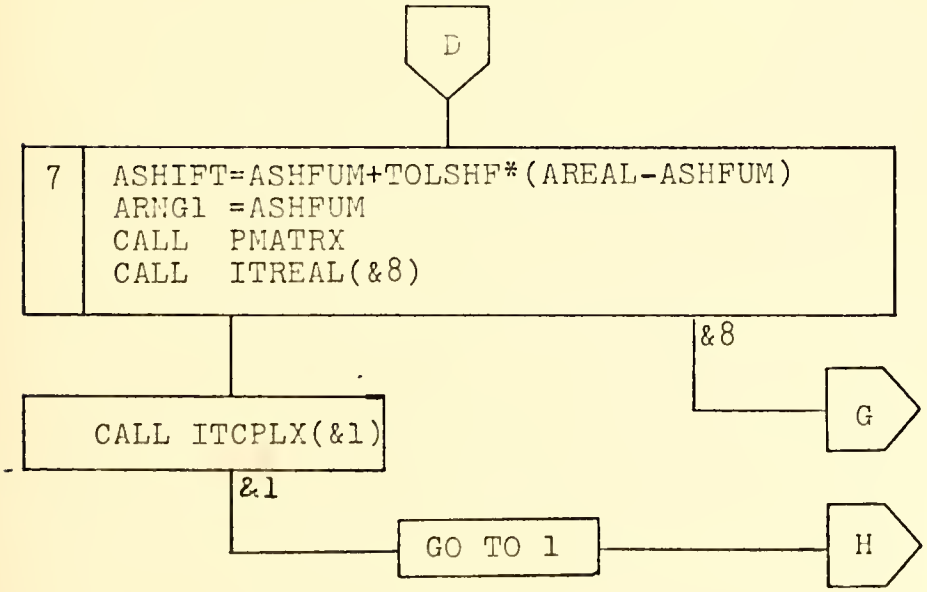


FIGURE 15 continued

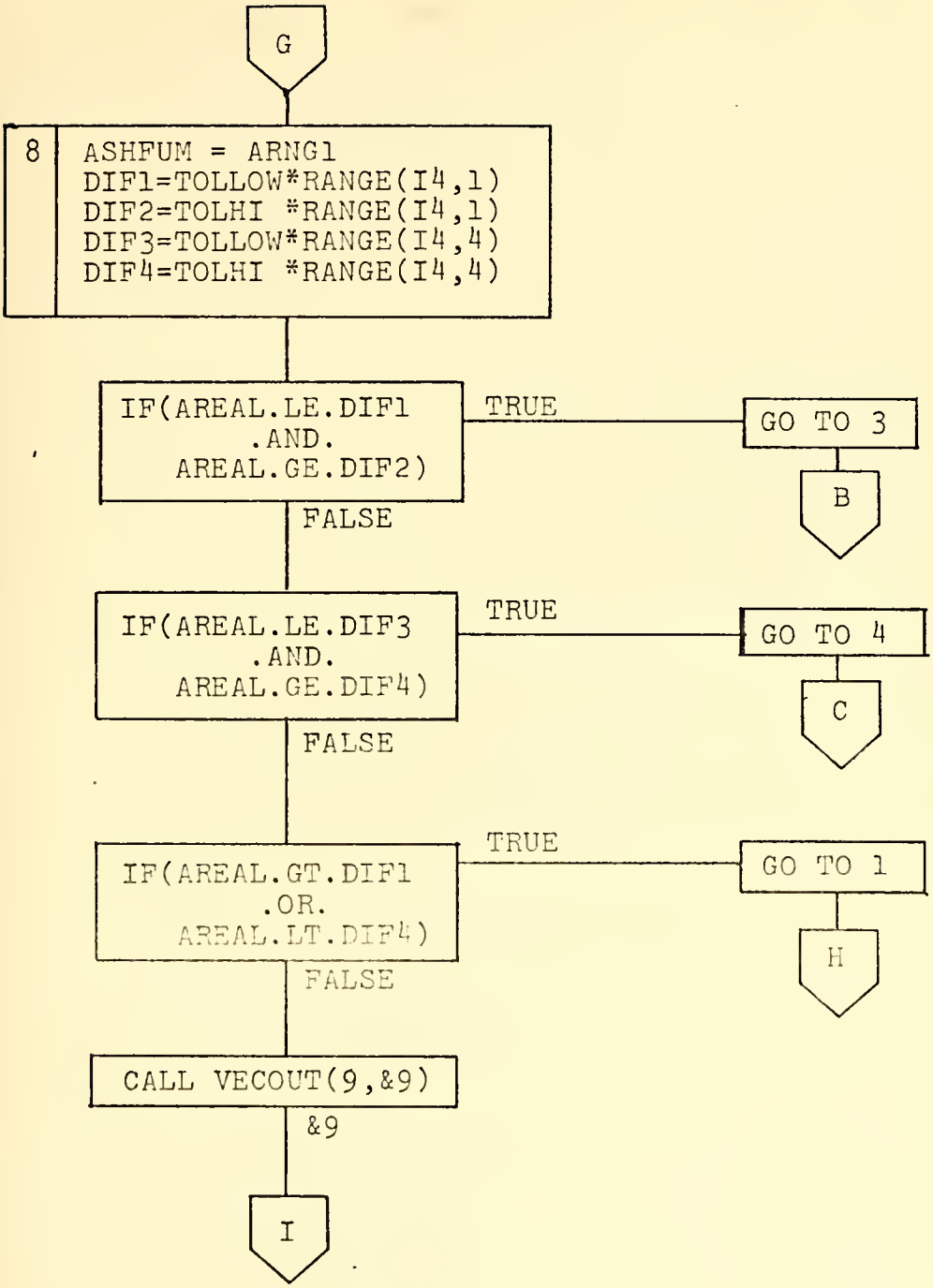


FIGURE 15 continued

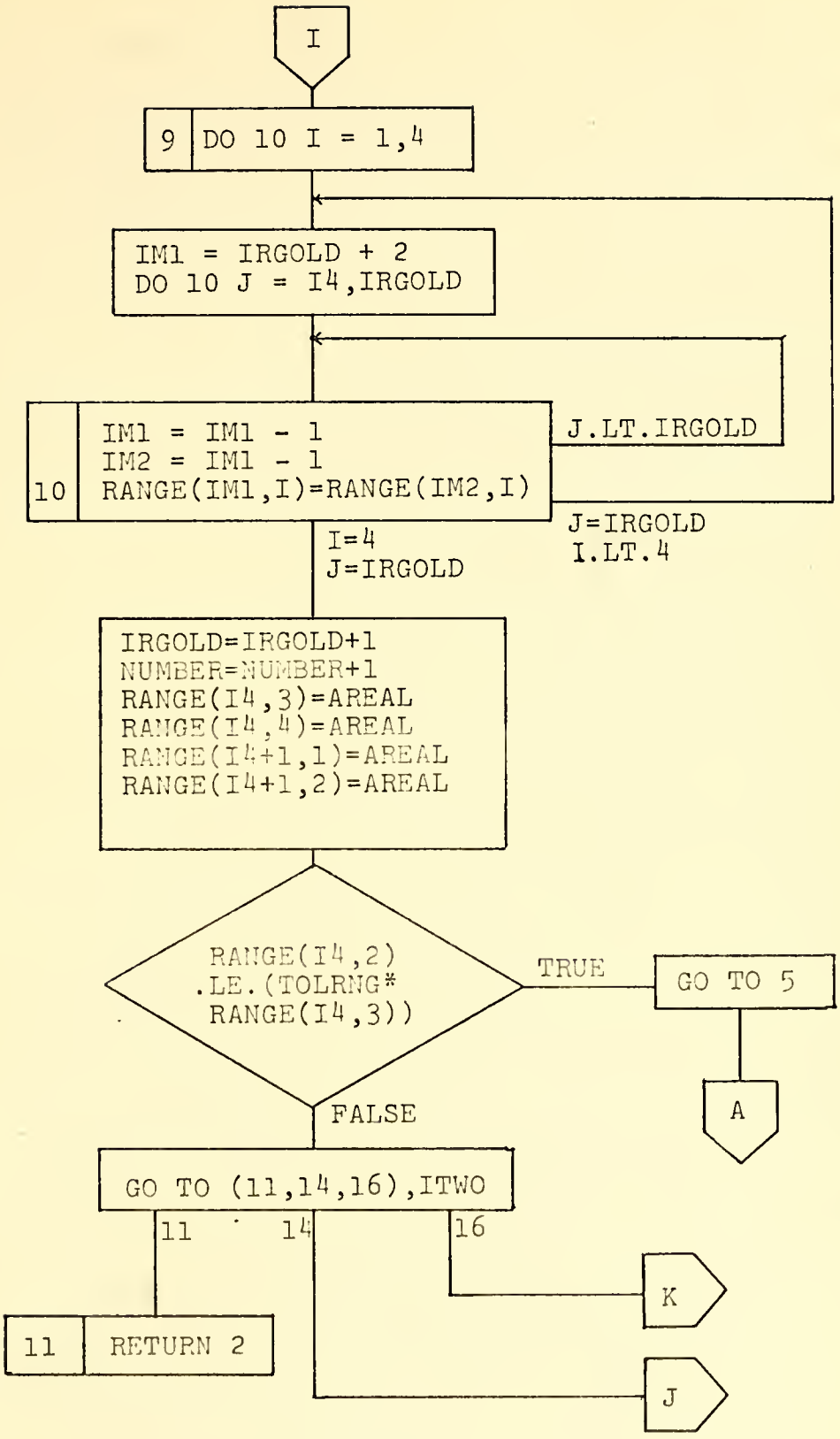


FIGURE 15 continued

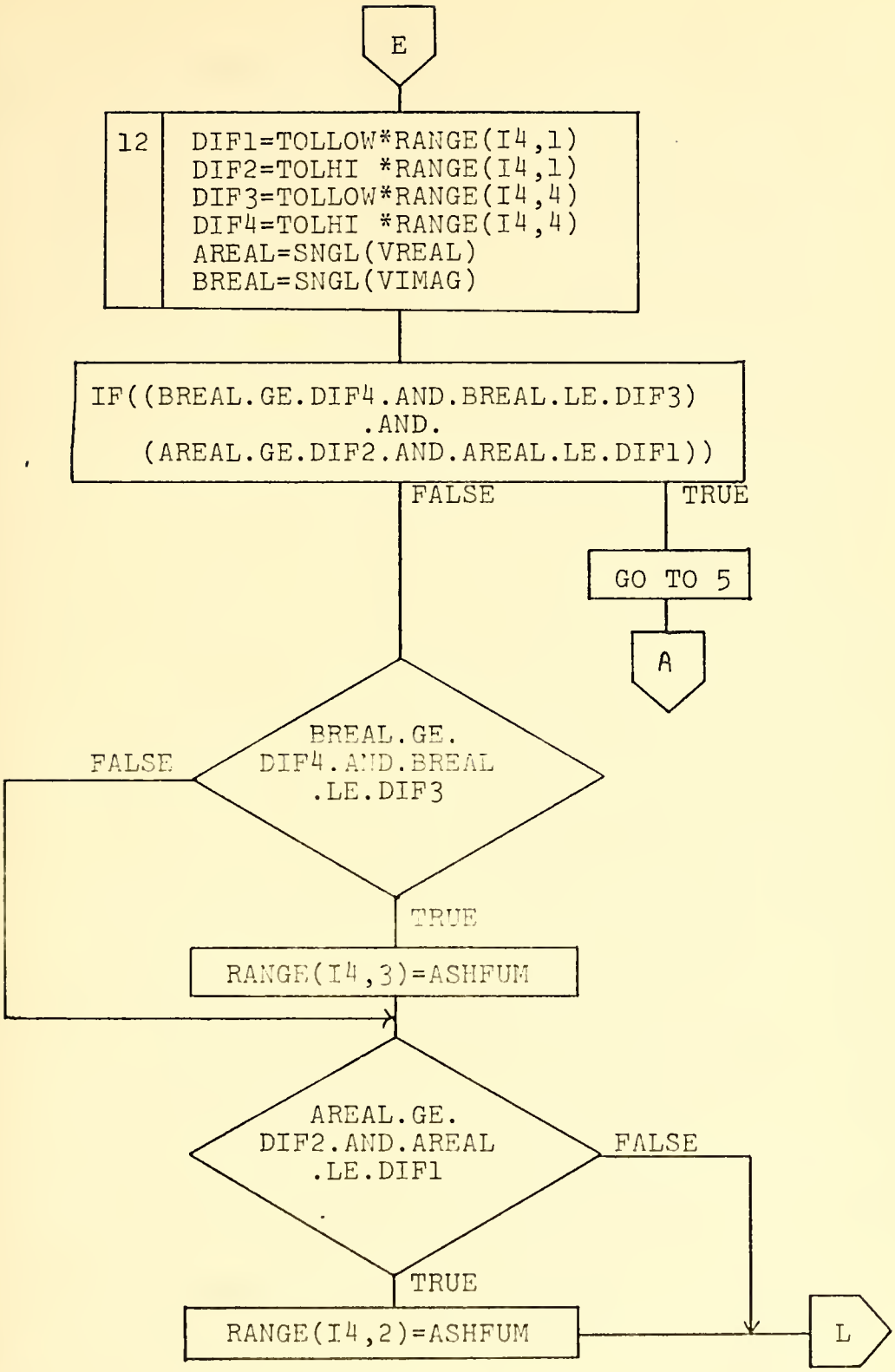


FIGURE 15 continued

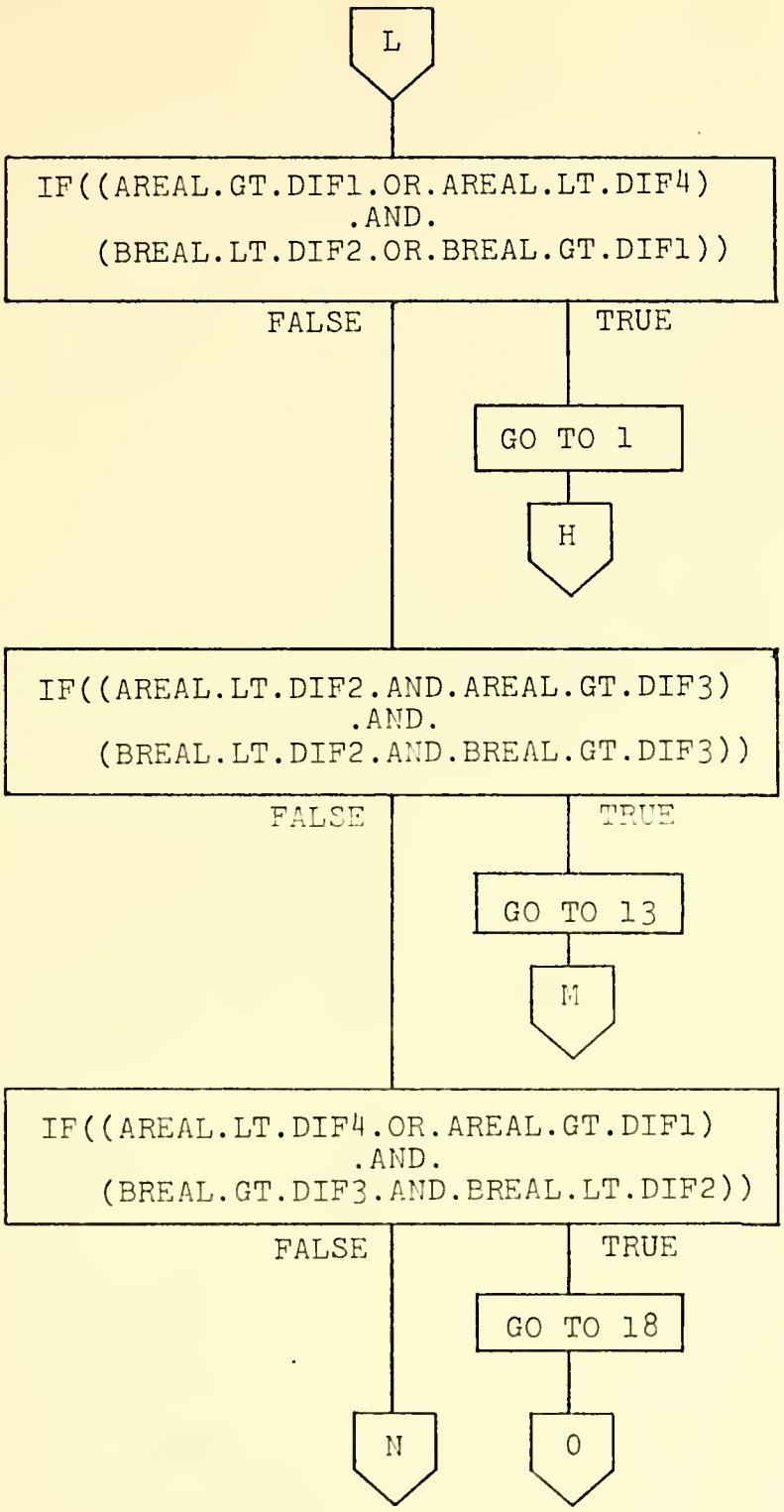


FIGURE 15 continued

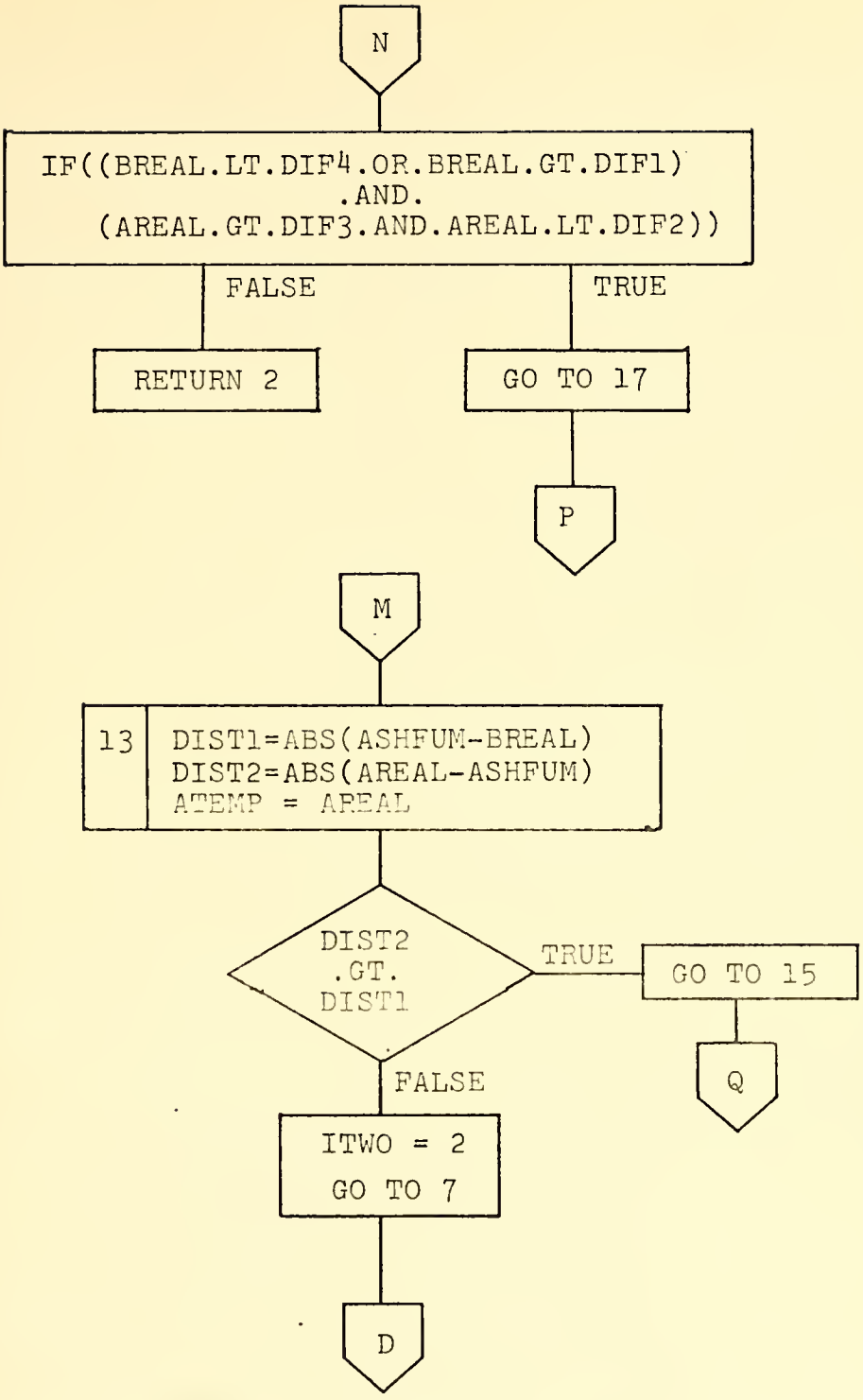


FIGURE 15 continued

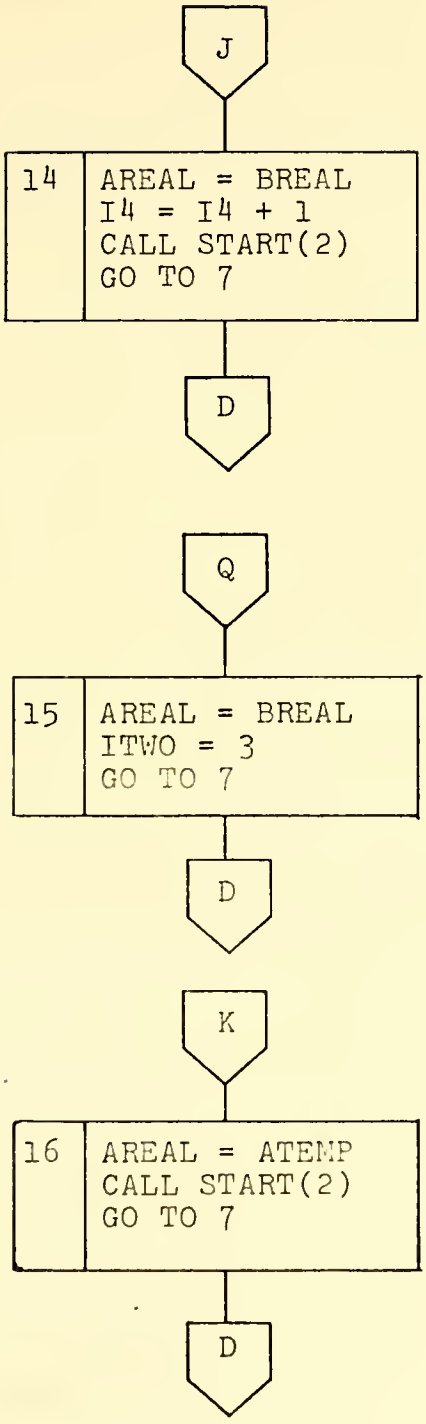


FIGURE 15 continued

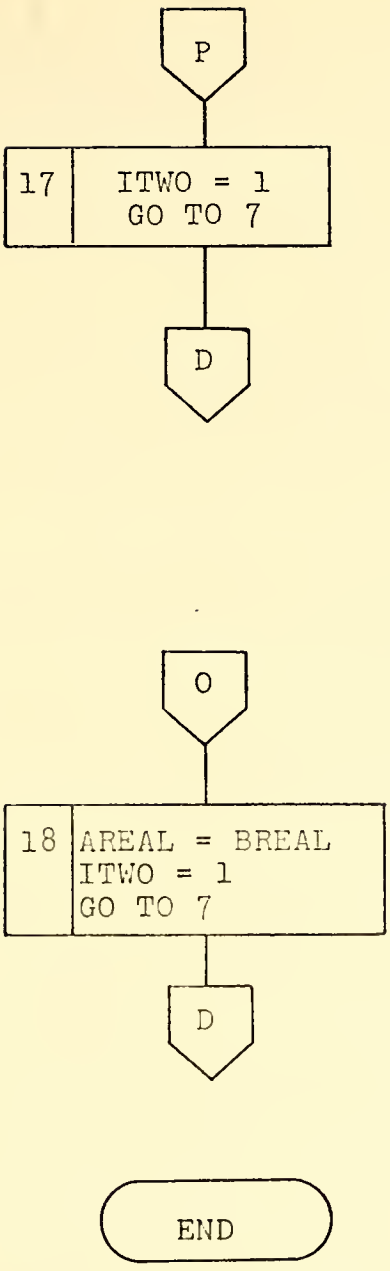


FIGURE 15 continued

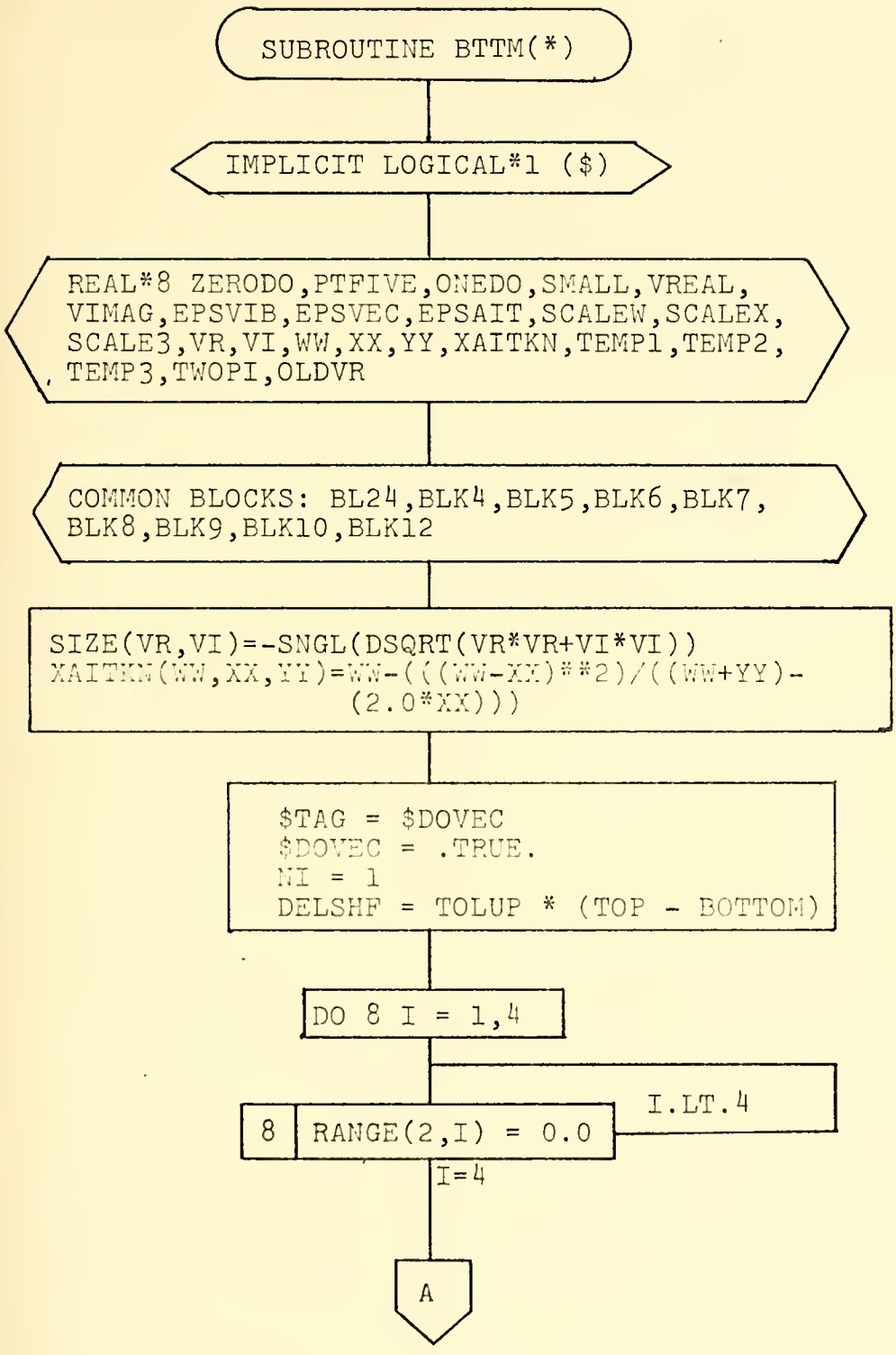


FIGURE 16 SUBROUTINE BTTM(*) FLOWCHART

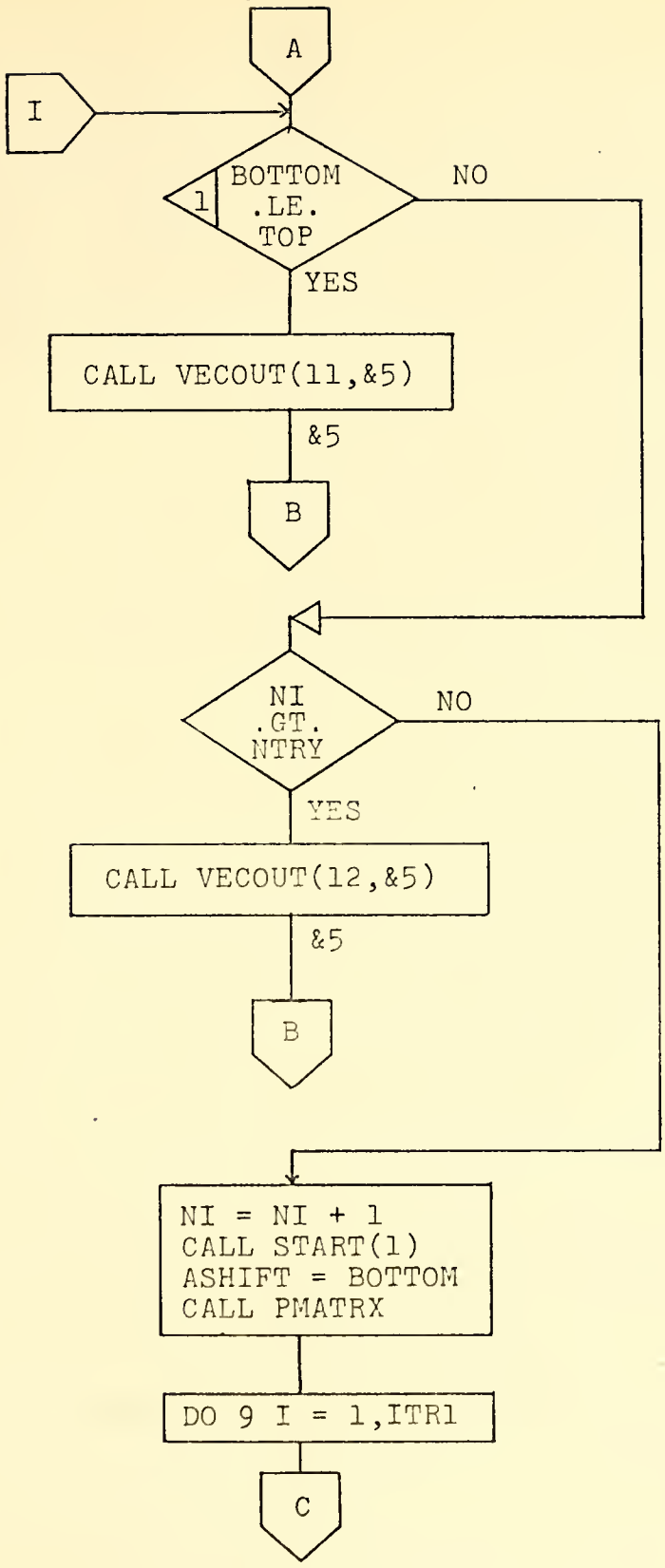


FIGURE 16 continued

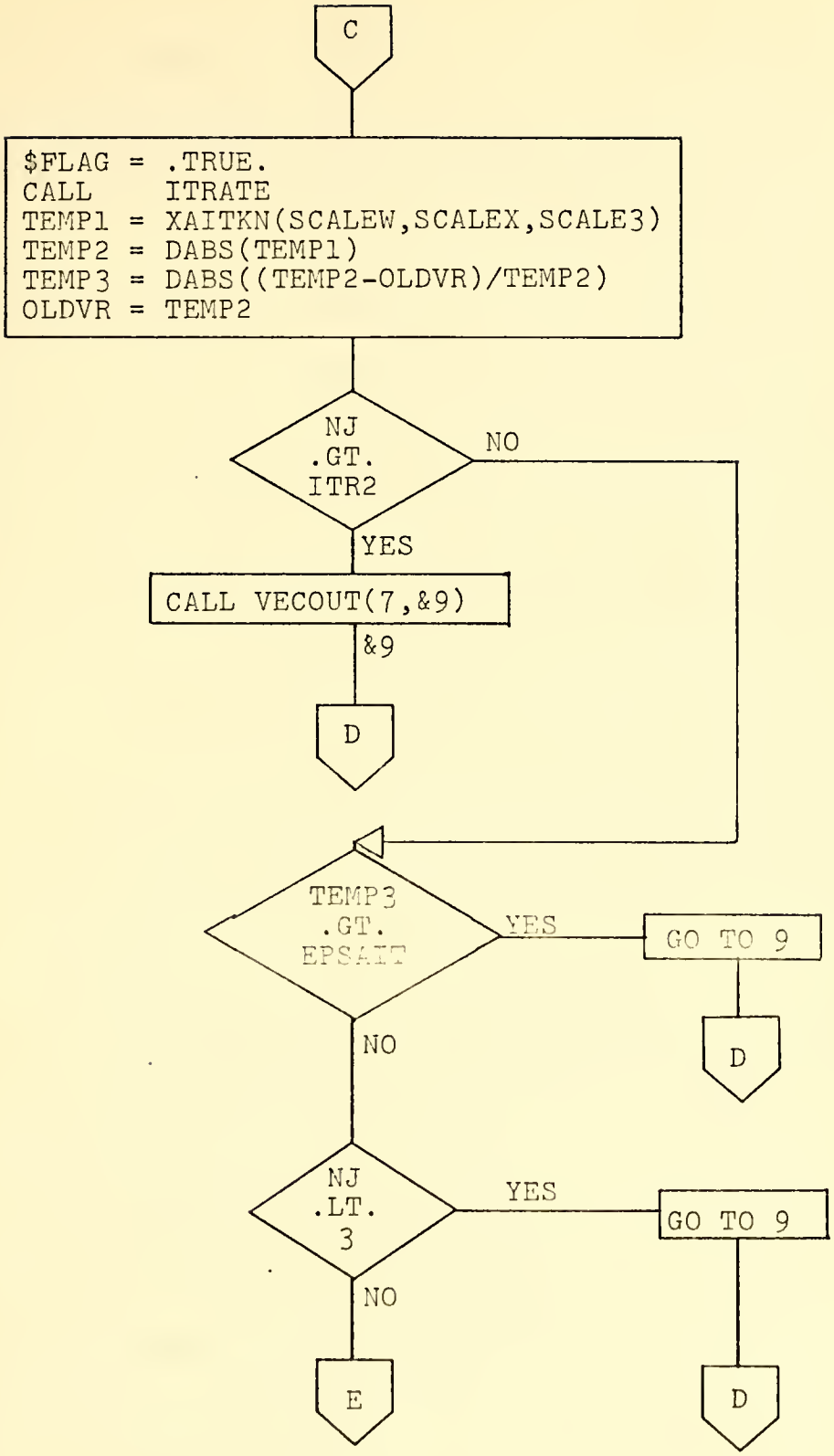


FIGURE 16 continued

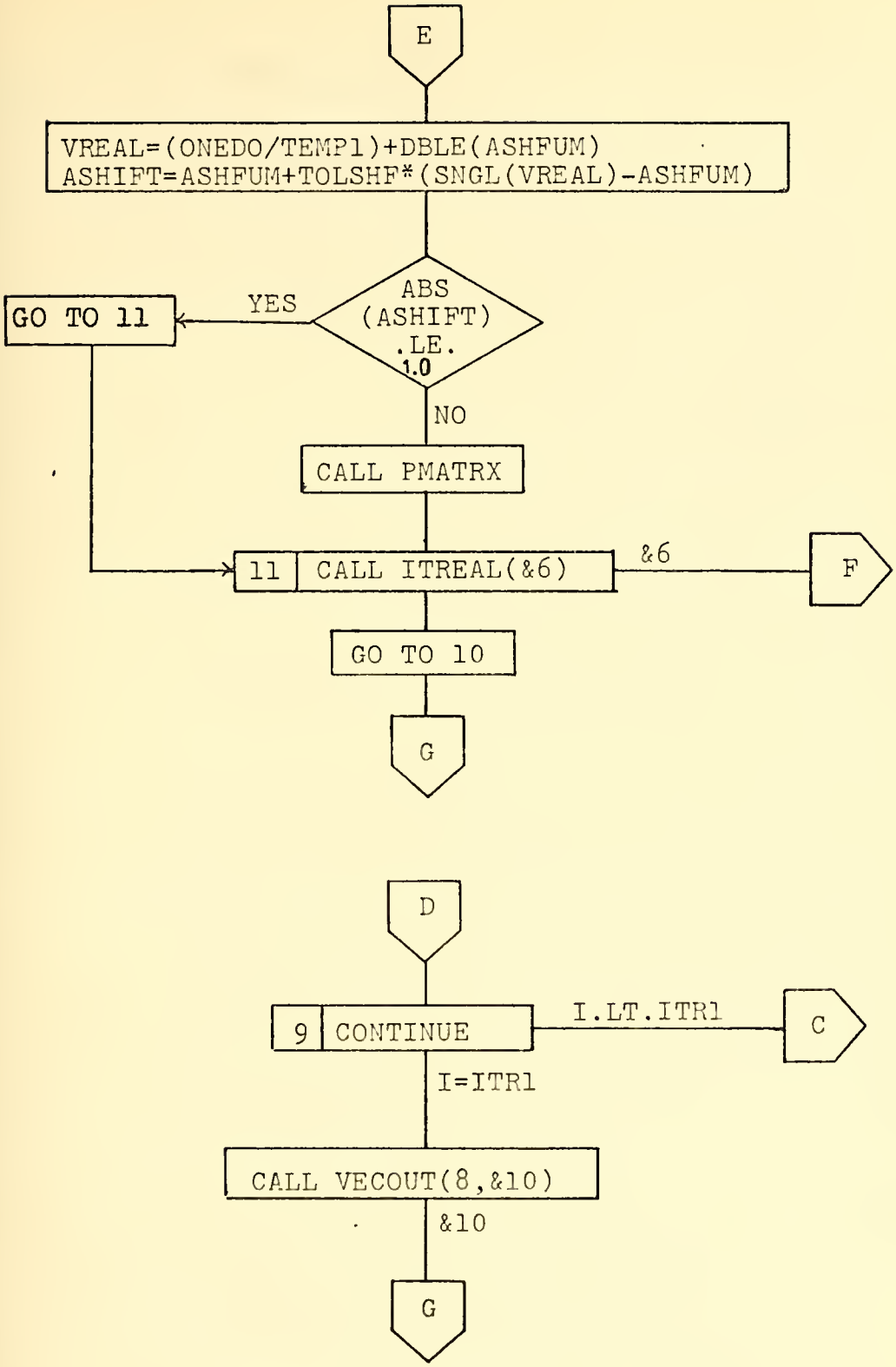


FIGURE 16 continued

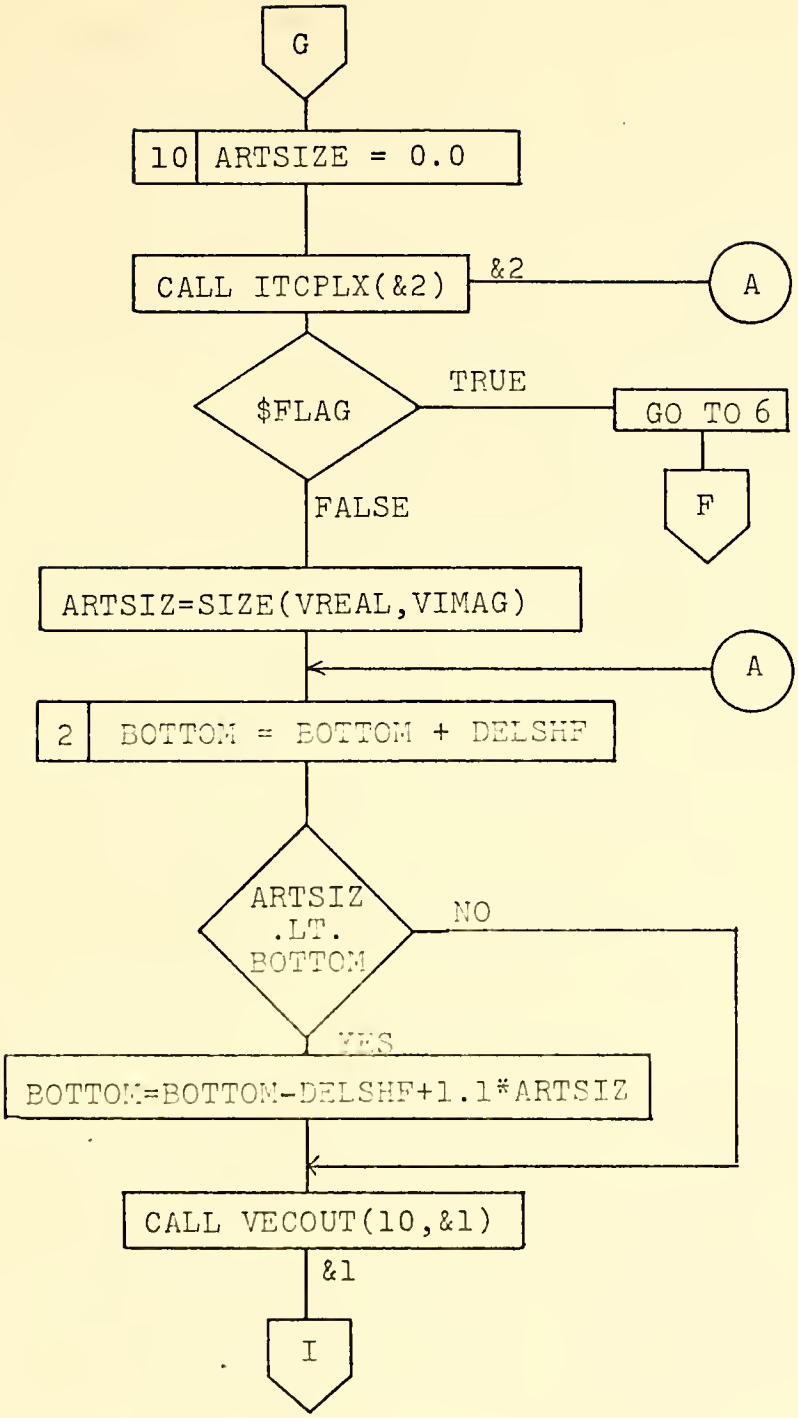


FIGURE 16 continued

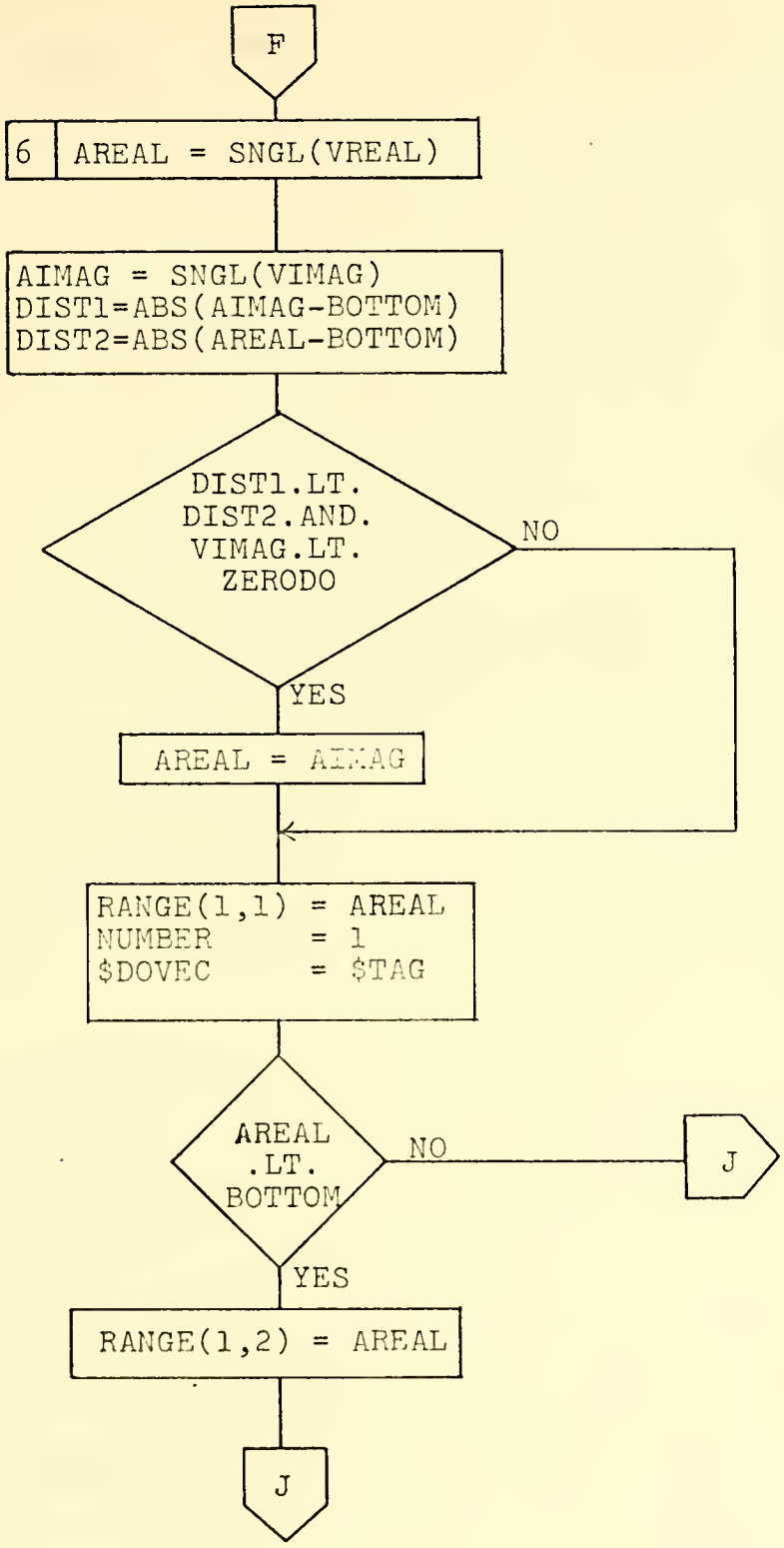


FIGURE 16 continued

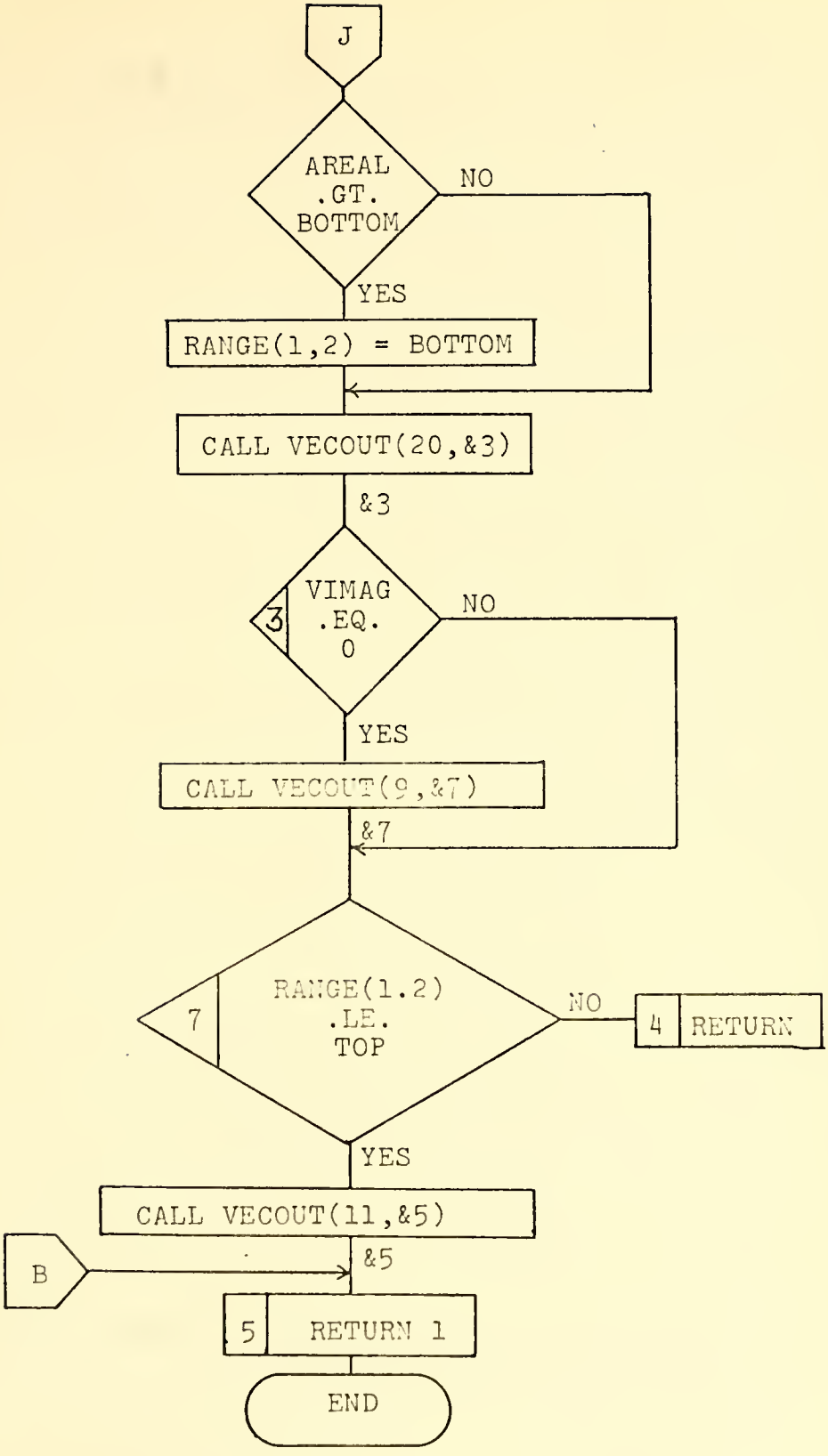


FIGURE 16 continued

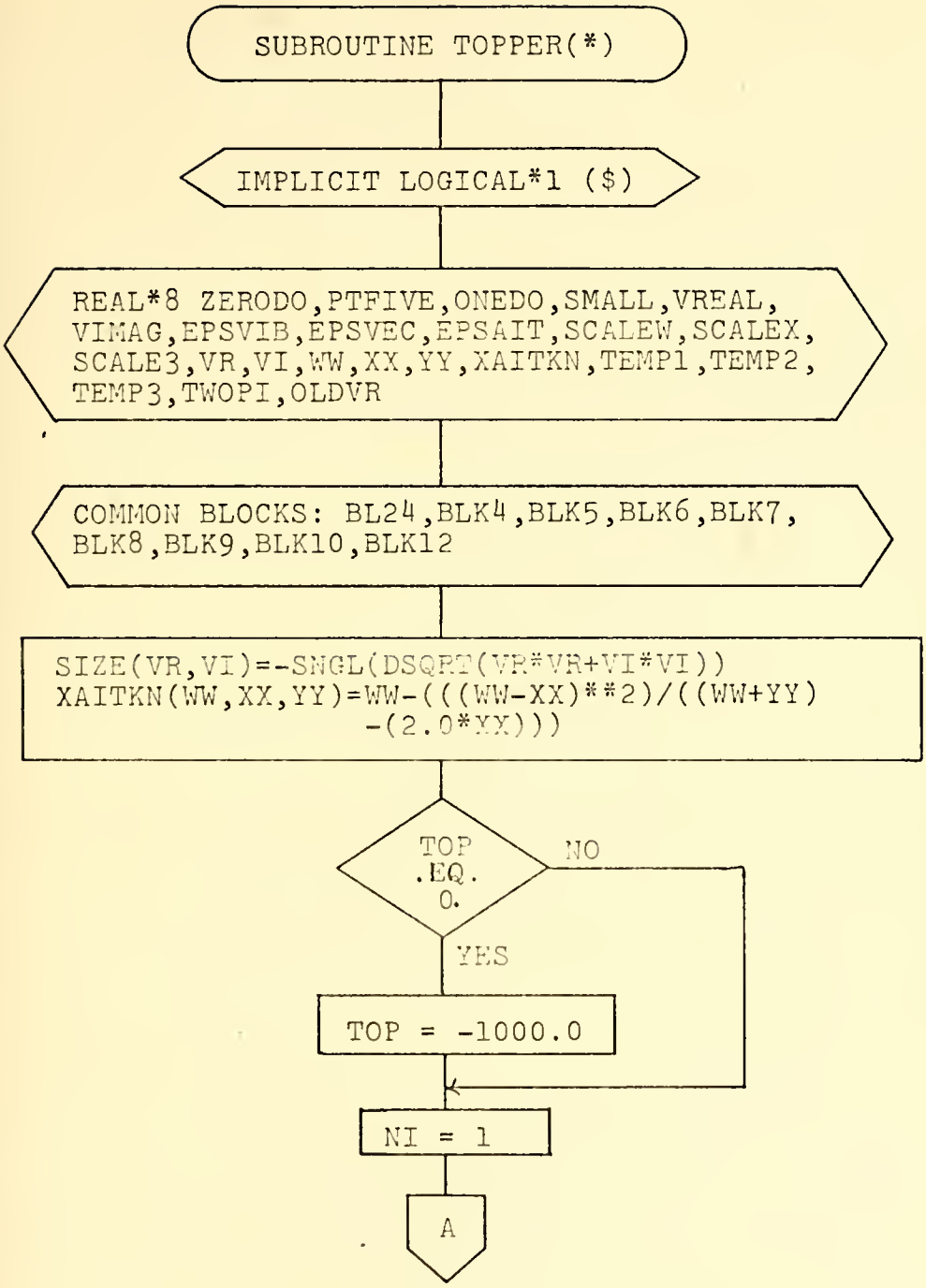


FIGURE 17. SUBROUTINE TOPPER(*) FLOWCHART

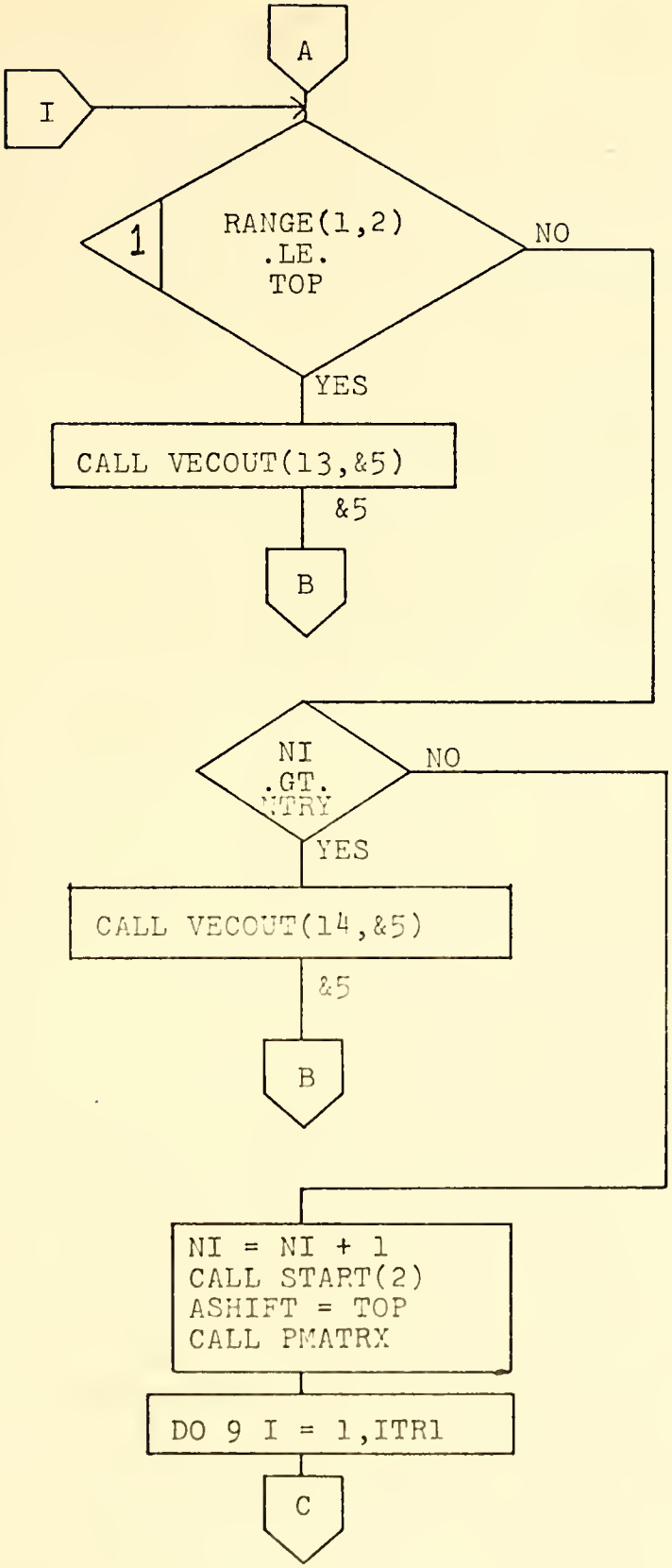


FIGURE 17 continued

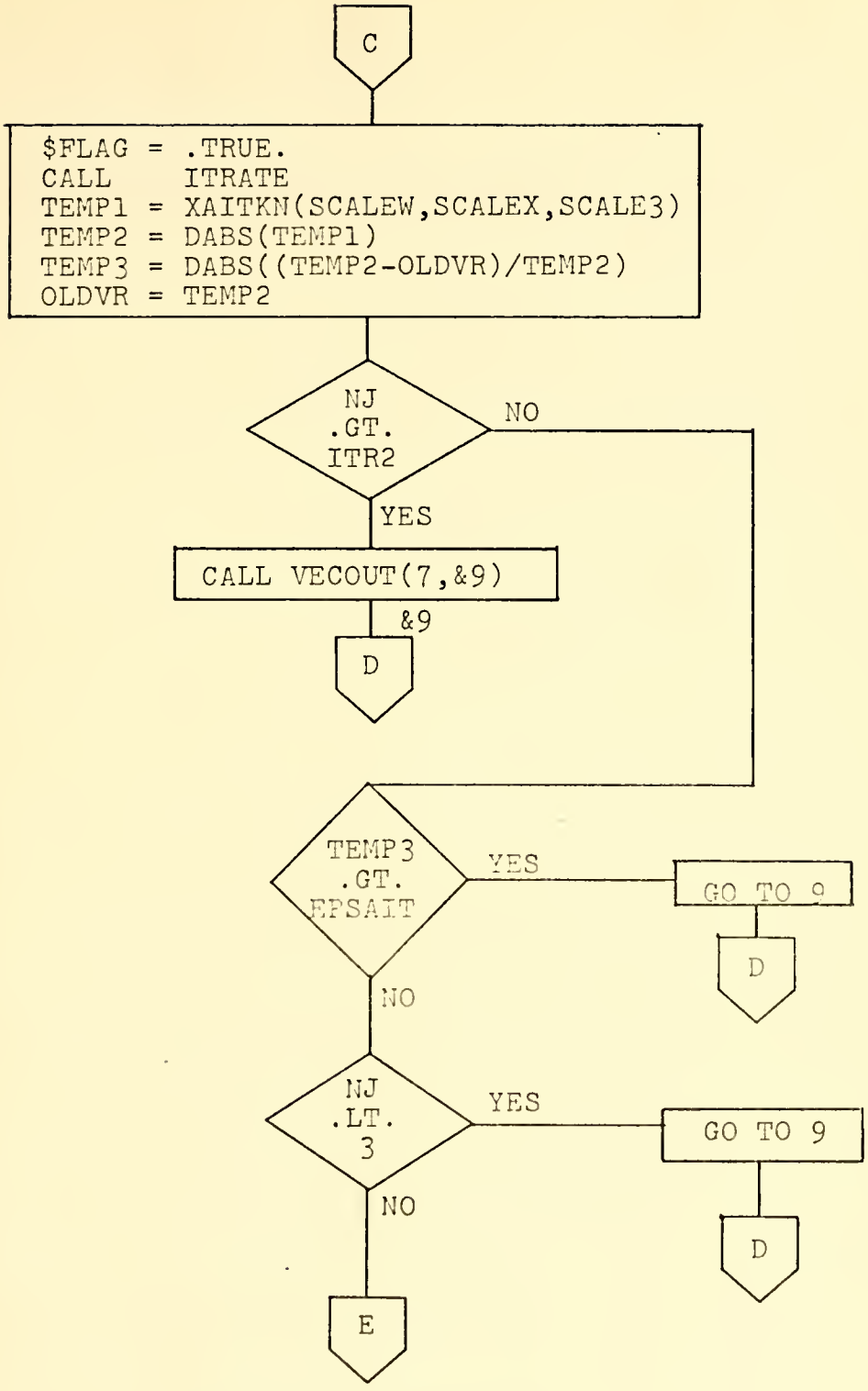


FIGURE 17 continued

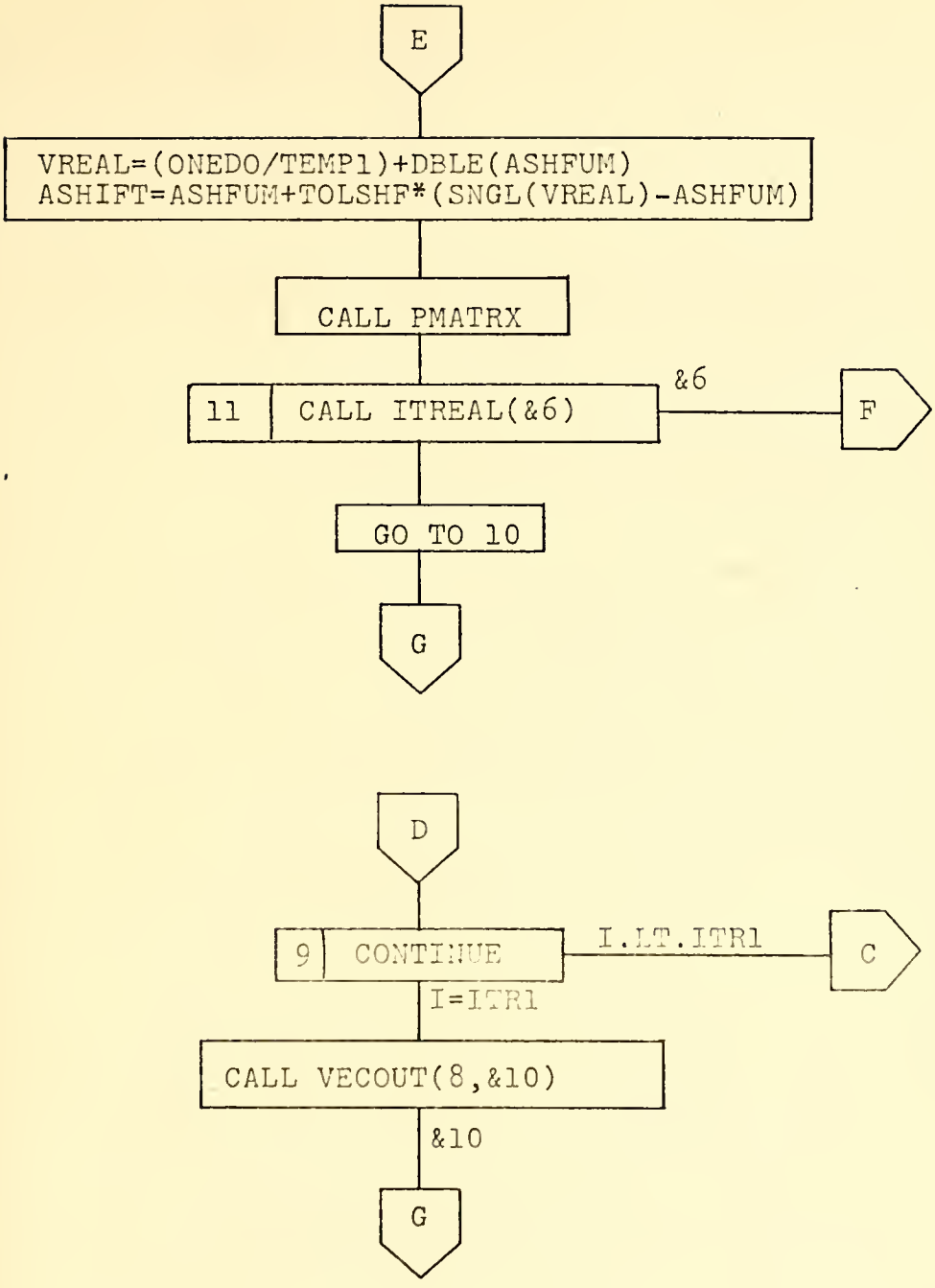


FIGURE 17 continued

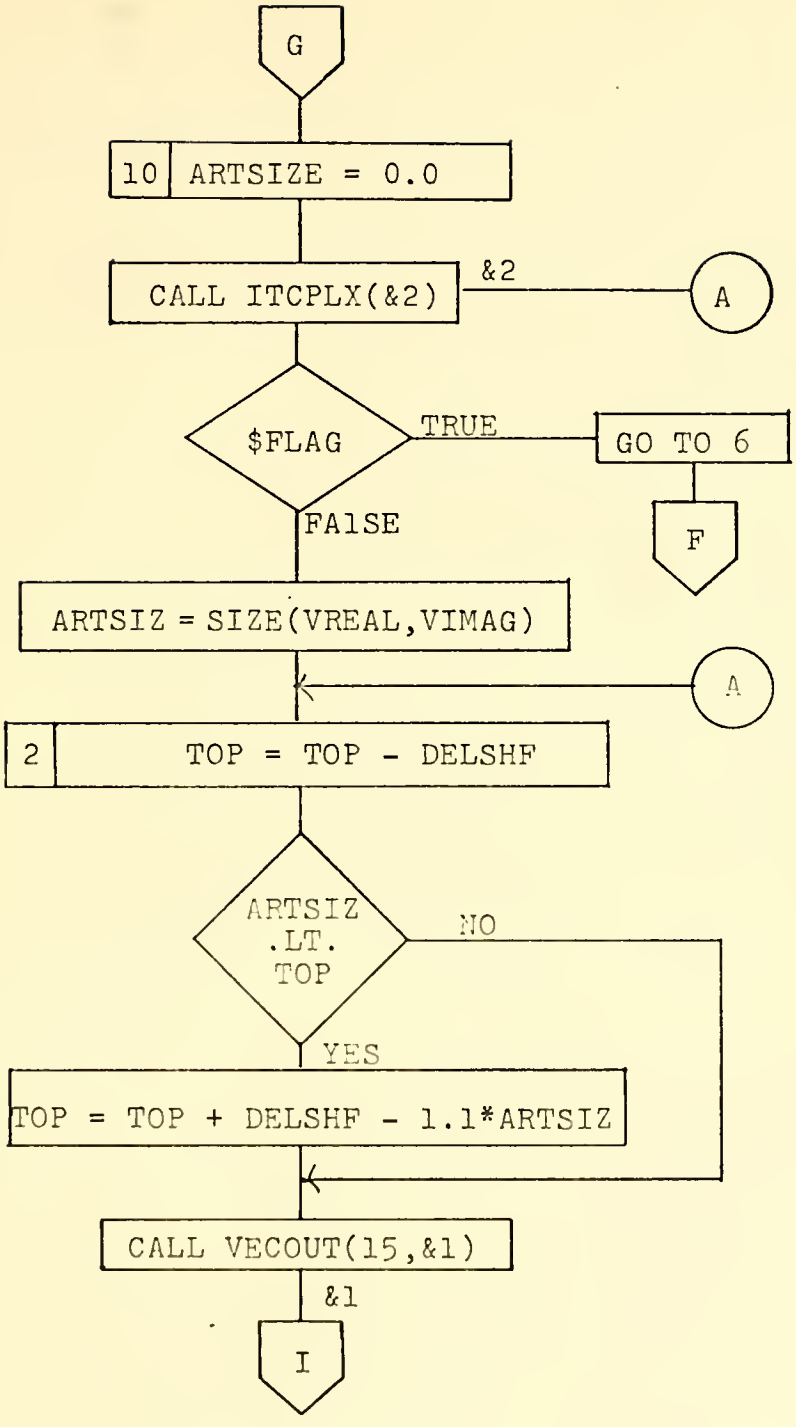


FIGURE 17 continued

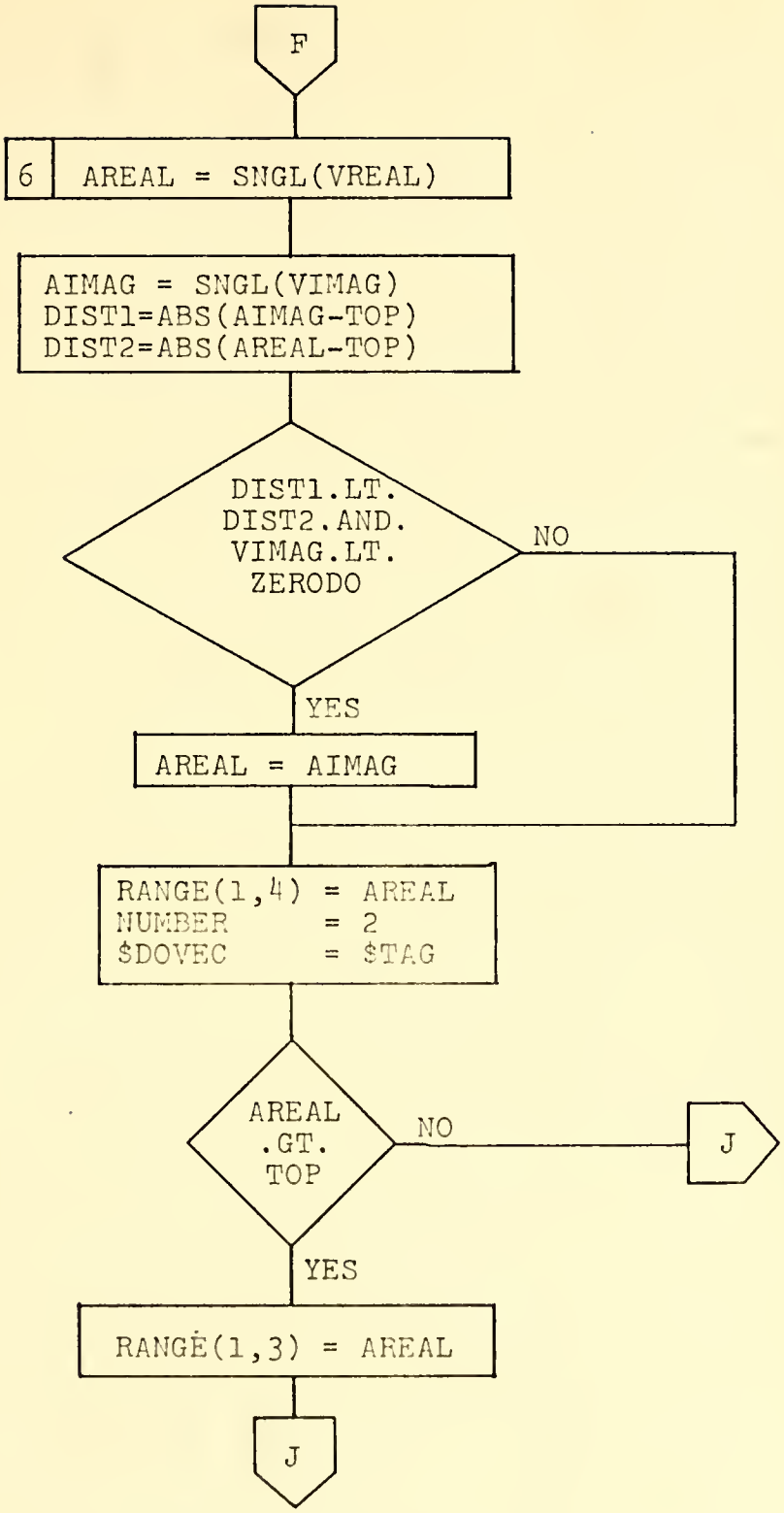


FIGURE 17 continued

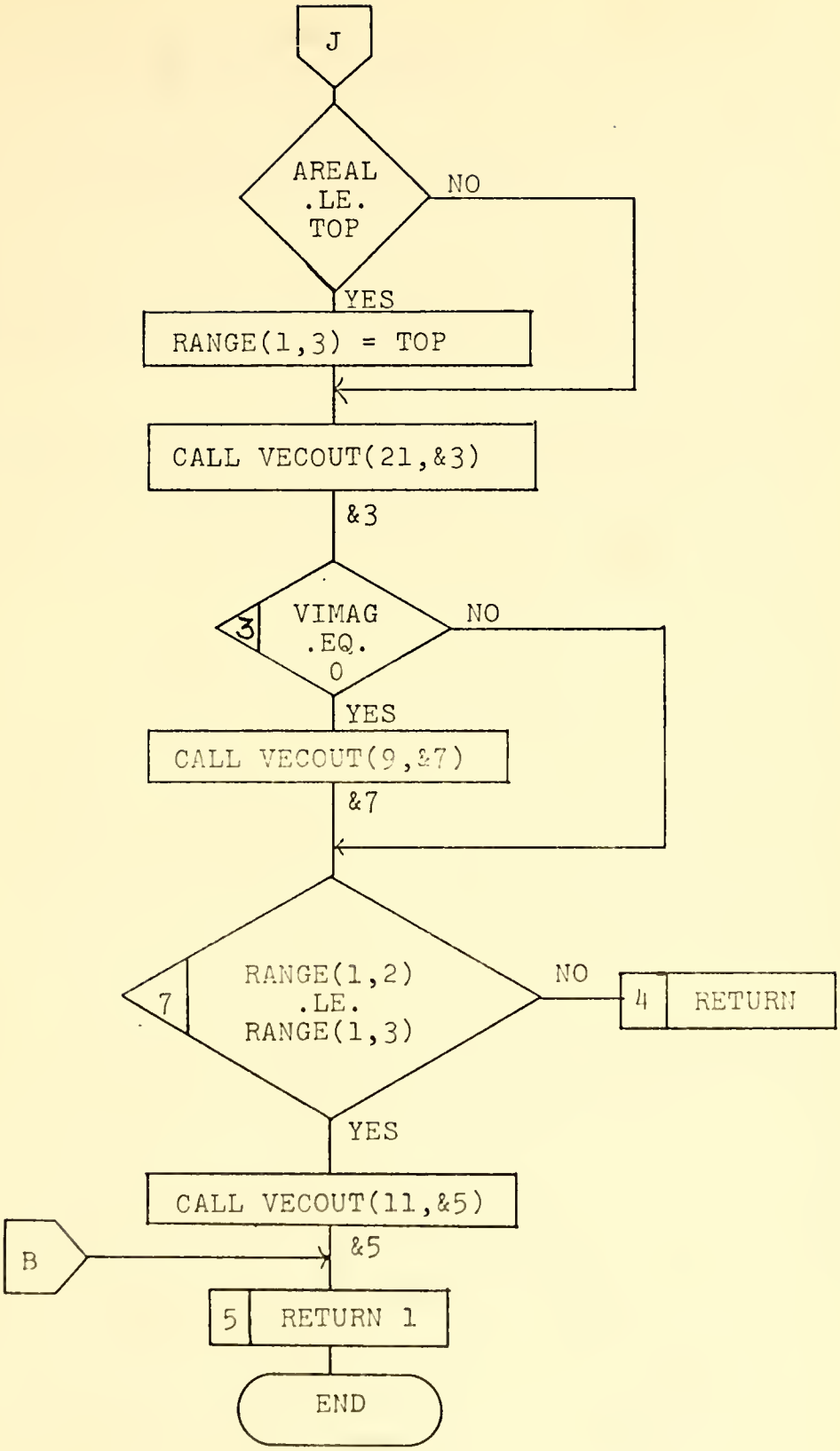


FIGURE 17 continued

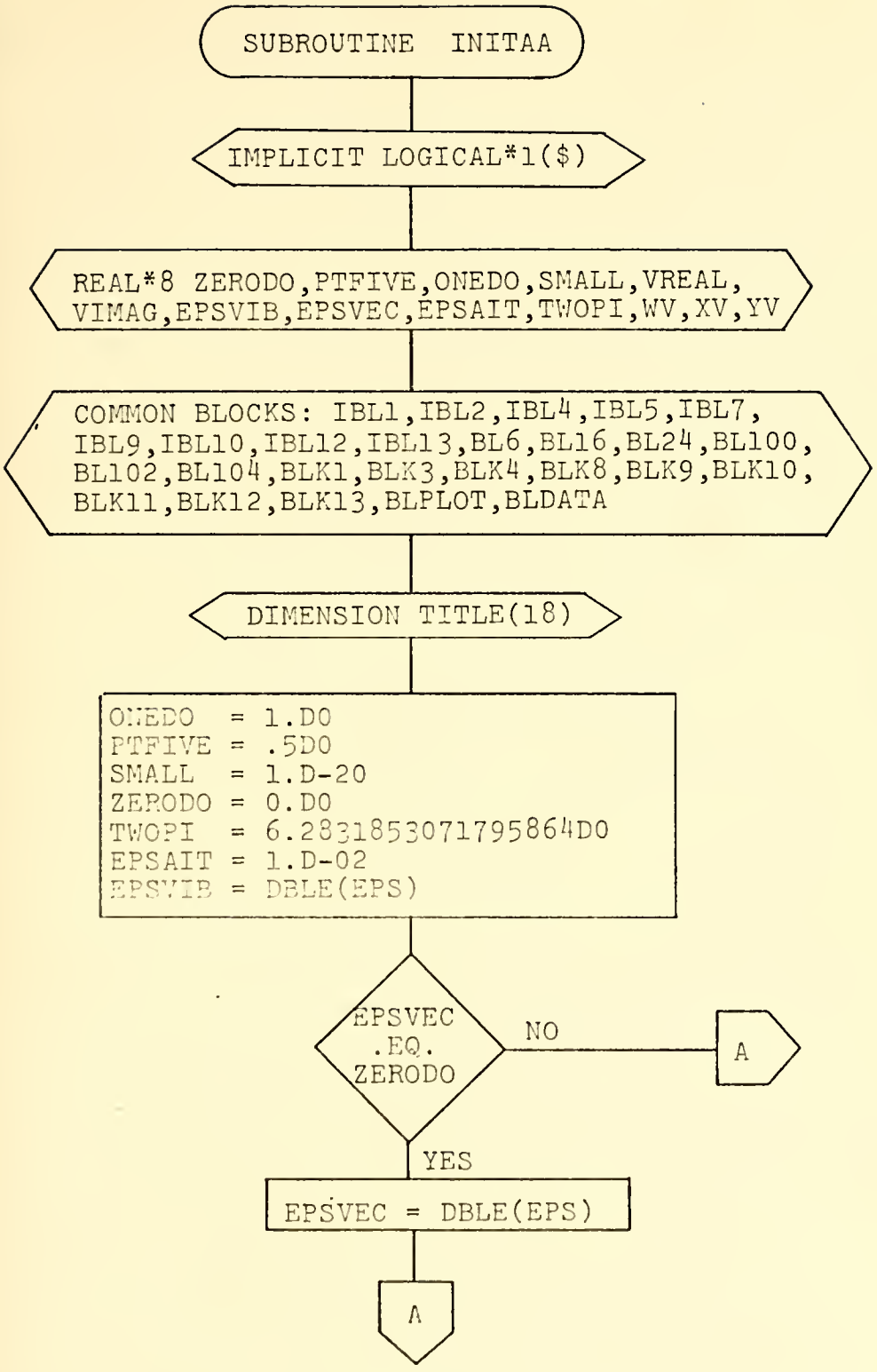


FIGURE 18. SUBROUTINE INITAA FLOWCHART

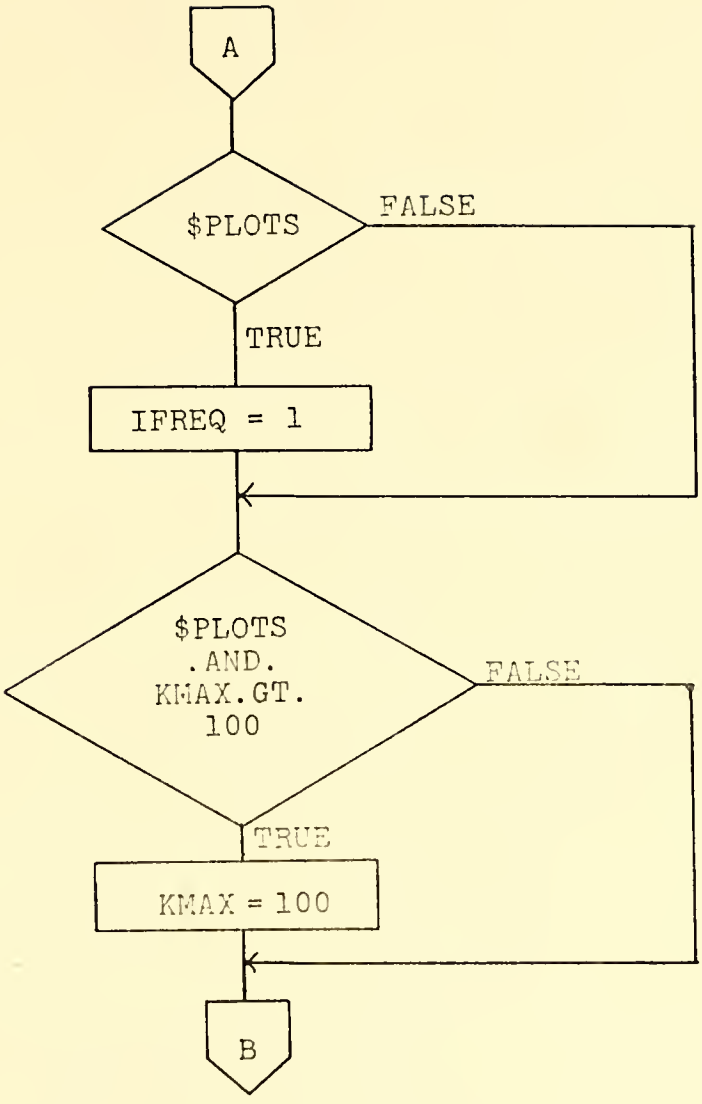


FIGURE 18 continued

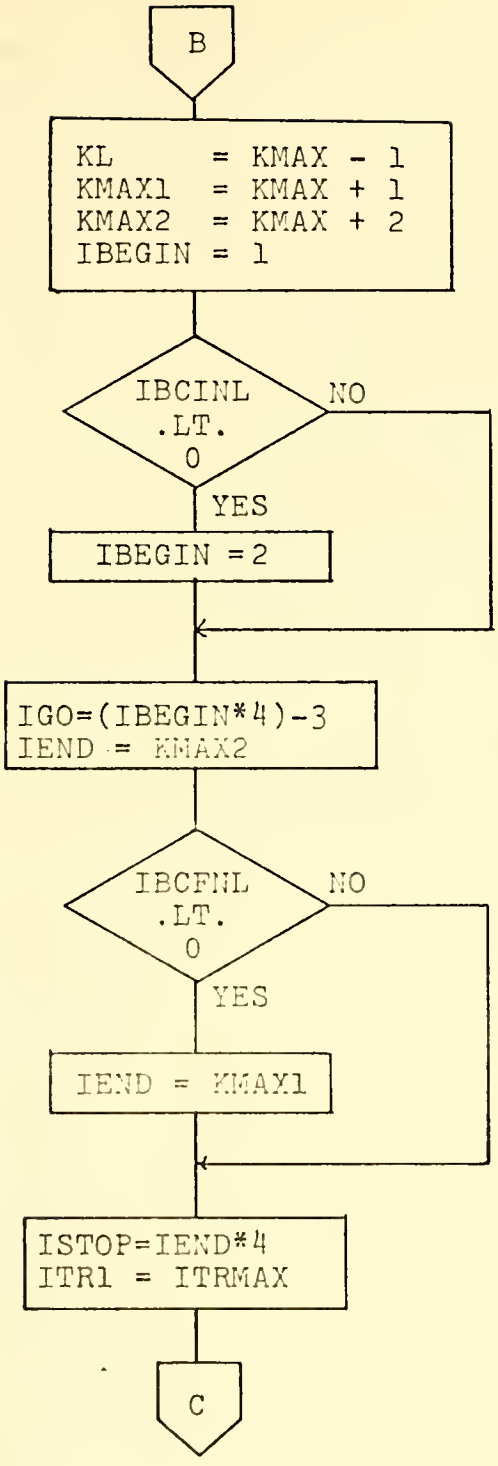


FIGURE 18 continued

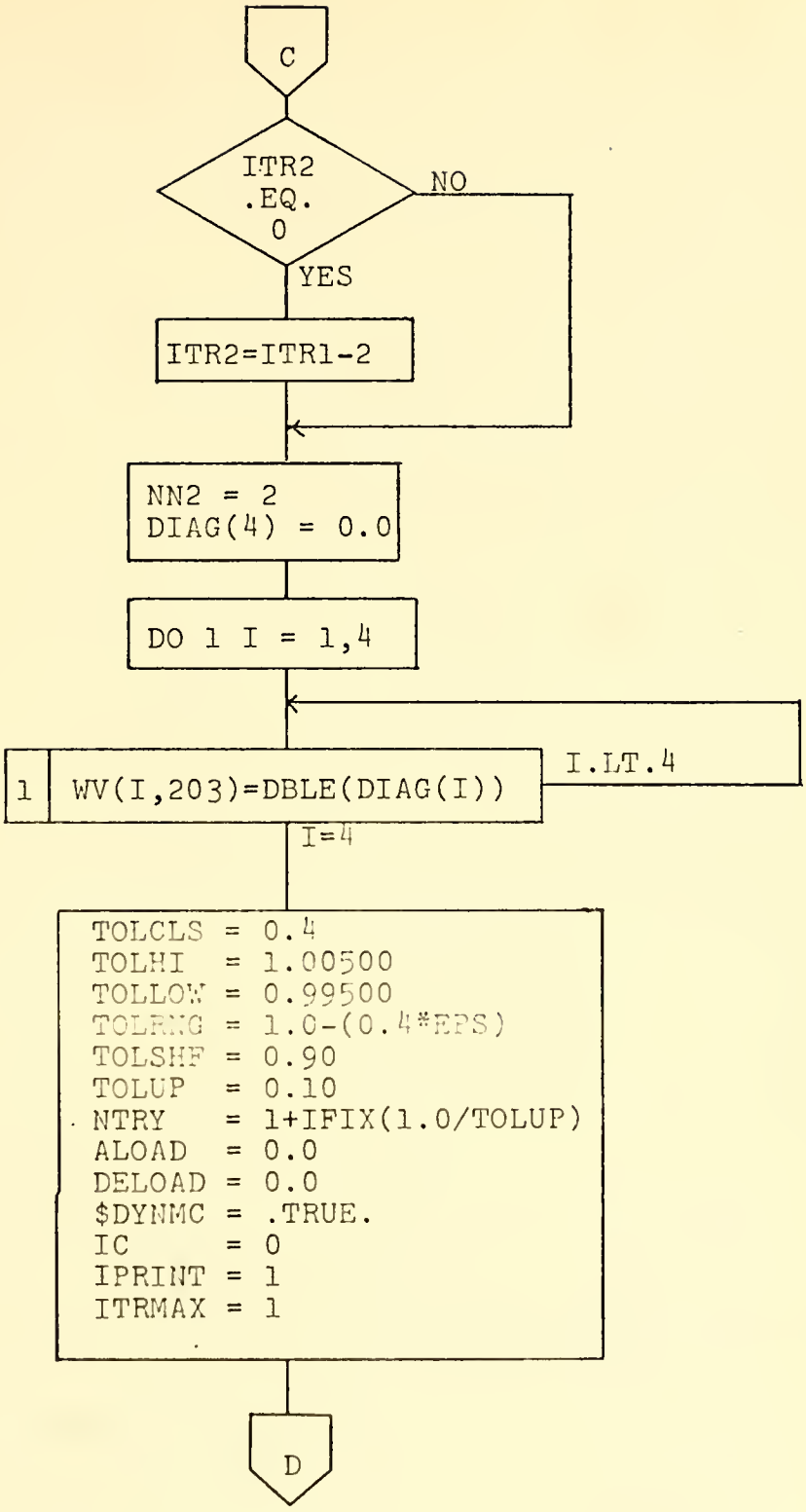


FIGURE 18 continued

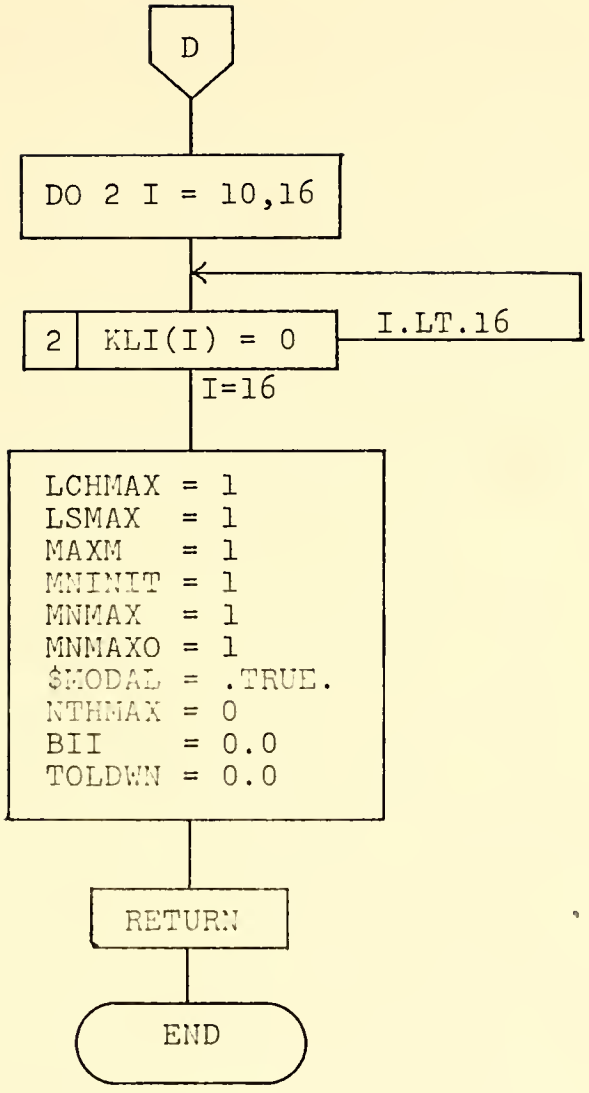


FIGURE 18 continued

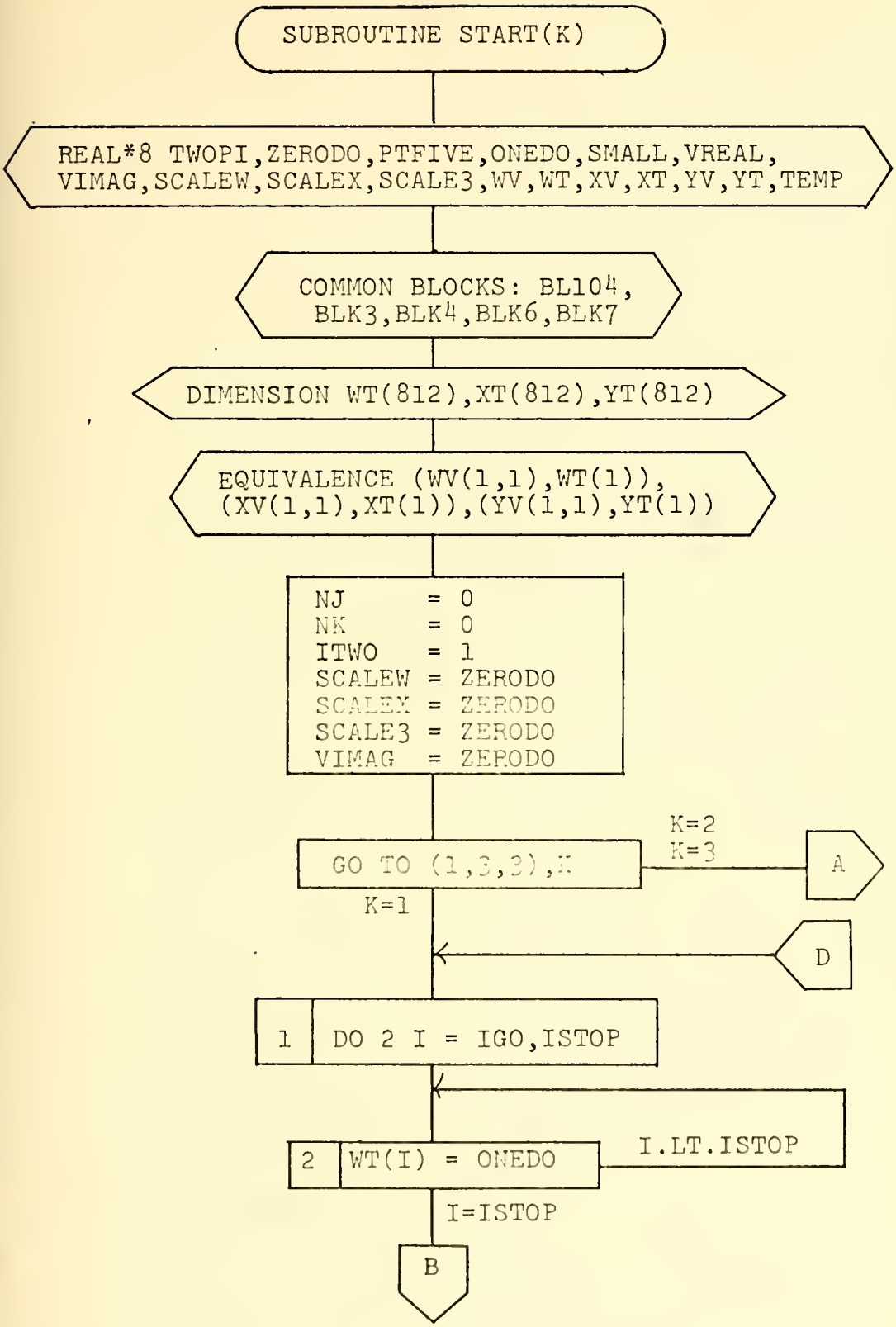


FIGURE 19. SUBROUTINE START(K) FLOWCHART

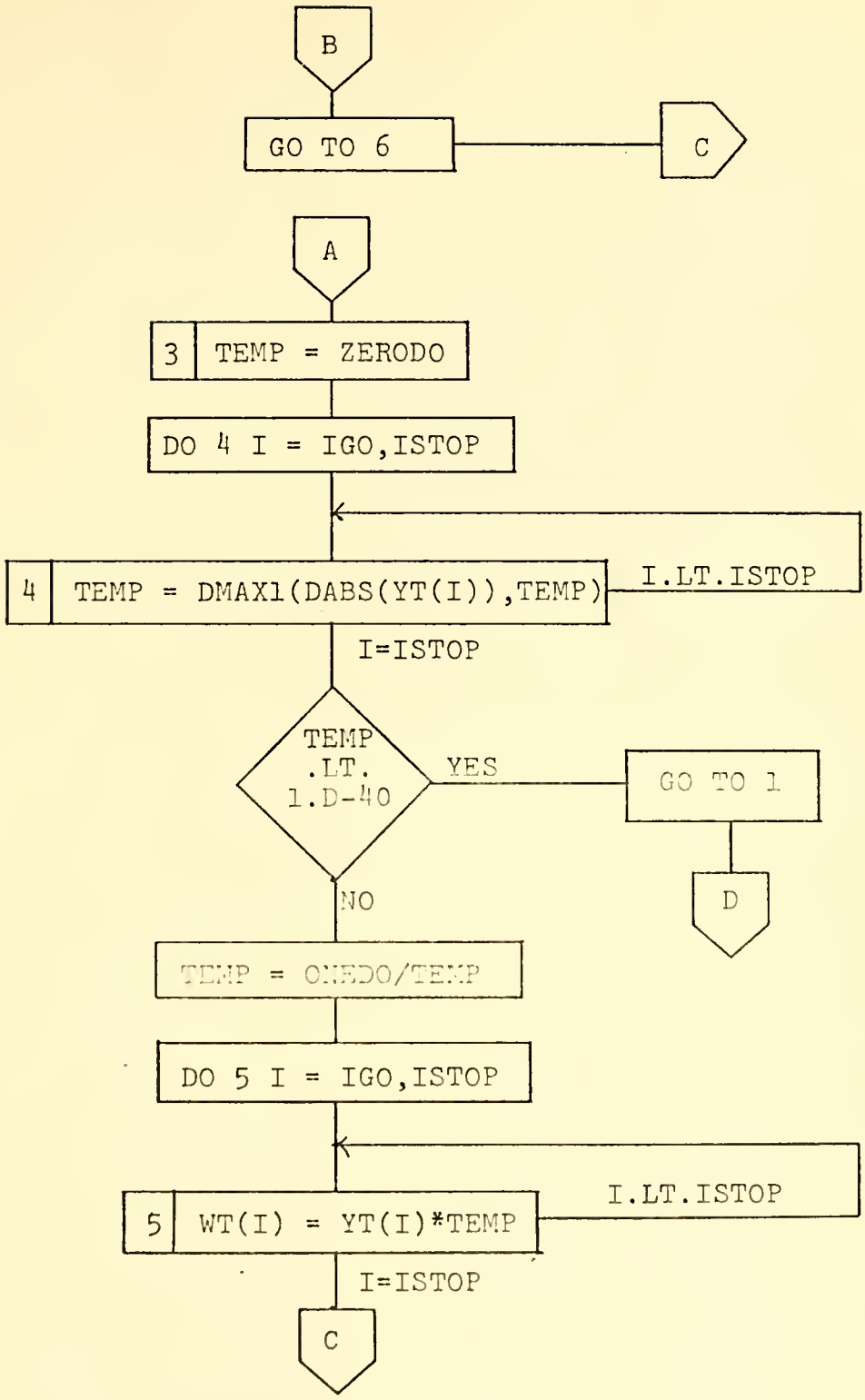


FIGURE 19 continued

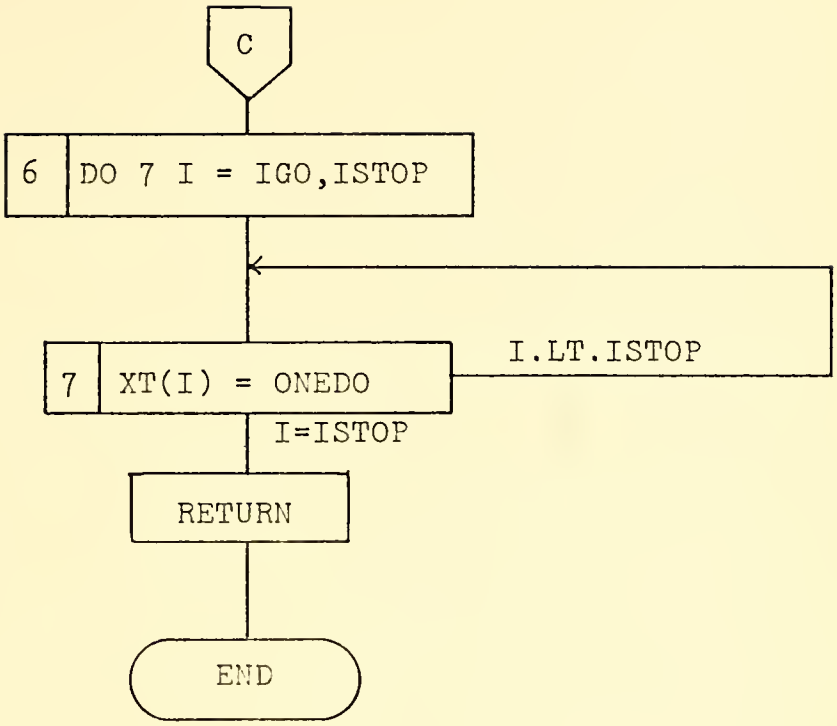


FIGURE 19 continued

```
 THE LOWEST FREQUENCY IN HARMONIC 3 IS DETERMINED

 DIAGONAL OF MASS MATRIX (MU-BAR) IS ( 1. 1. 1. 0. )

 THE CONVERGENCE TEST ON THE VECTOR WILL BE PERFORMED
       CONVERGENCE CRITERION IS  0.10D-01

       CONVERGED SOLUTION. ITERATIONS =     18

RAYLEIGH QUOTIENT FREQUENCY =    0.40103772723439D 03 HZ

       ( SOLUTION VECTOR PRINTED HERE )

 3 SHIFT POINT(S) FOR HARMONIC 5 IS(ARE) PROVIDED

 DIAGONAL OF MASS MATRIX (MU-BAR) IS ( 0. 0. 1. 0. )

THE CONVERGENCE TEST ON THE VECTOR WILL NOT BE PERFORMED

       CONVERGED SOLUTION. ITERATIONS =     7

RAYLEIGH QUOTIENT FREQUENCY =    0.43982589633472D 02 HZ

       ( OTHER SOLUTIONS FOLLOW )
```

FIGURE 20. Subroutine VECOUT(IND,*) Print-out Samples

```
THIS IS A FREQUENCY RANGE SEARCH FOR HARMONIC 3

BOTTOM OF RANGE IS  0.7000E 01 HZ          TOP OF RANGE IS  0.1200E 02 HZ
        BOTTOM =  0.4900E 02                        TOP =  0.8400E 02

      DIAGONAL OF MASS MATRIX (MU-BAR) IS ( 1. 1. 1. 0. )

        BOTTOM =  0.4900E 02
        LOWEST EIGENVALUE (RANGE(1,1)) = 0.5300E 02
        RANGE(1,2) = 0.5300E 02

        CONVERGED SOLUTION. ITERATIONS =    9

    RAYLEIGH QUOTIENT FREQUENCY =   0.8358963820957D 01 HZ

       ( SOLUTION VECTOR PRINTED HERE )

        TOP =  0.8400E 02
        HIGHEST EIGENVALUE (RANGE(1,4)) = 0.7000E 02
        RANGE(1,3) = 0.7000E 02

        CONVERGED SOLUTION. ITERATIONS =   11

    RAYLEIGH QUOTIENT FREQUENCY =   0.1000000000000E 02 HZ

         ( OTHER SOLUTIONS FOLLOW )


        5 FREQUENCIES HAVE BEEN FOUND
```

FIGURE 20. continued

Diagnostic Messages:

EPS IS VERY SMALL (0.10D-09)

CHECK PERMISSIBLE NUMBER OF ITERATIONS (3)

NO CONVERGENCE TO A REAL EIGENVALUE AFTER 99 ITERATIONS
RAYLEIGH QUOTIENT FREQUENCY = 0.123456789101111D 04 HZ
{THE FREQUENCY IS IMAGINARY}
THE LAST, NEXT TO LAST AND NEXT TO NEXT TO LAST EIGENVALUES ARE
-0.987654321098D 02 -0.987654321005D 02 -0.987656789876D 02

NO CONVERGENCE AFTER 99 ITERATIONS
EIGENVALU(S) ARE 0.123456D 01 0.123456D-01
THE LAST, NEXT TO LAST AND NEXT TO NEXT TO LAST VECTOR ELEMENT MAXIMUMS ARE
0.123456789102D 03 -0.987654323456D-04 0.123456789345D 04

CONVERGED SOLUTION - TWO REAL EIGENVALUES
AN INCREASE IN THE CONVERGENCE CRITERIA (EPS) AND/OR AN INCREASE IN THE PERMISSIBLE NUMBER OF ITERATIONS(ITRMAX) IS RECOMMENDED AS A POSSIBLE SOLUTION TO THE PROBLEM OF A POORLY CONDITIONED MATRIX.
ITERATIONS ARE 99 FREQUENCIES ARE 0.123456D 02 0.123456D-02
{THE FIRST FREQUENCY IS IMAGINARY
BOTH FREQUENCIES ARE IMAGINARY}

CONVERGED SOLUTION - COMPLEX EIGENVALUE
ITERATIONS ARE 99 EIGENVALUE IS 0.123456D 02 0.123456D-02

THE RANGE SIZE HAS BEEN REDUCED
BOTTOM OF RANGE IS NOW 0.123456E 01

UNABLE TO DETERMINE A MINIMUM FREQUENCY AFTER 10 TRIES

THE RANGE SIZE HAS BEEN REDUCED
TOP OF RANGE IS NOW 0.123456E 04

UNABLE TO DETERMINE A MAXIMUM FREQUENCY AFTER 10 TRIES

RANGE HAS BEEN COVERED

FIGURE 20. continued

APPENDIX D

SOME EXPERIENCE AND RECOMMENDATIONS ON USING SATANS-III

Selecting the program mode which determines the minimum frequency for a given Fourier harmonic (circumferential wave number), KEY1 = 0, causes the vector test to be performed. To determine the minimum eigenvalue without performing the vector test, the supplied shift point program mode should be used, KEY1 = 1, with the shift point set to zero and KEY3 = 0.

The vector convergence test is intended for use only when there is some difficulty experienced in extracting a particular mode.

Three shift points per Fourier harmonic may be supplied by the user, KEY1 = 3, on one card. The starting vector for the first value is all ones. The starting vector for the one or two subsequent values is determined by the pseudo sweeping technique developed for subroutine RANGER. However, if one of those values is zero, the starting vector will be all ones. Three shift points enable a difficult mode to extract to be bracketed.

In the frequency range search operation the user supplies the upper and lower bounds of the range. In the lower modes the iteration process is sensitive to the choice of starting vectors. On infrequent occasions temporary convergence to one of two closely spaced eigenvalues

might cause a mode to be missed. This may be avoided by making the convergence criterion very tight, using the vector test and using a large number of iterations. This is a high price to pay for the average case and the supplied shift point mode of the program is a more economical approach and the recommended course of action. The simultaneous determination of the mode shapes provides an easy method of spotting if and which mode is missing.

Because the starting vectors are biased toward the lower frequencies the range bounds supplied by the user may not be the range actually searched by SATANS-III. The range actually searhced is from BOTTOM to TOP as subsequently determined by subroutine RANGER. If, for example, a user specifies the range 1000 to 2000 Hz the bias toward lower frequencies might cause the actual search range to be from 800 to 1500 Hz. For this reason it is recommended that at least the upper bound be over-estimated to insure the complete search of the desired range. A good rule of thumb for the lower frequencies is an over-estimation of 50 to 100%. If a range is very small it might be searched more economically by the supplied shift method.

The only restriction on output is that only modal data is available. Since only one harmonic at a time is involved this was not thought to be a severe restriction. No matter what the input value is NTHMAX is subsequently set to zero.

In plotting output it must be remembered that KMAX must be less than 101.

Some difficulty has been encountered in nondimensionalizing and it is recommended that it not be employed. To avoid this difficulty set SIGO, ELAST, TKN and CHAR (input card three) to 1.0. Adjustments in the relative magnitude of the shift point values may be made using TEEO.

It was discovered that using the Naval Postgraduate School's IBM-360 under OS/MVT, Release 18, and the catalogued procedure FORTHCLG overflowed the linkage editor symbol table. This was corrected by the specification PARM.LINK ='SIZE=(150K,20480),LIST,MAP',REGION.LINK=200K

APPENDIX E

SATANS-III INPUT CARD DATA GUIDE

Card	Columns	Format	Item	Example	Meaning
1	1-72	18A4	TITLE	-	Enter any 72 characters
2	1- 5	I5	NO	1	The problem number, 0<NO<100000.
2	6- 8	L3	$DYNMC	F	$DYNMC = F for a static analysis.
				T	$DYNMC = T for dynamic or vibration analyses.
2	9-10	L2	$VIBES	F	$VIBES = F for dynamic and static analyses.
				T	$VIBES = T for a vibration analysis.
2	11-15	I5	IMODE	0	For no modal output data.
				1	For modal output data.
2	16-20	I5	NDIMEN	0	Dimensional output data.
				1	Nondimensional output data.
2	21-25	I5	NTHMAX	8	Summed solution will be printed at NTHMAX meridians, 0<=NTHMAX<=36. Not applicable in a vibration analysis.
2	26-30	I5	IFREQ	3	Solution will be printed at the first and last stations and every IFREQth intermediate station.
2	31-35	I5	IPRINT	2	Every IPRINTth converged solution will be printed. Not applicable in a vibration analysis.

Card	Columns	Format	Item	Example	Meaning
2	36-40	I5	IBCINL	-1	If the shell has a pole at the first station.
				0	If the shell does not have a pole at the first station.
2	41-45	I5	IBCFNL	-1	If the shell has a pole at the last station.
				0	If the shell does not have a pole at the last station.
2	46-50	I5	KMAX	25	The number of meridional stations. KMAX<201 If plots are desired KMAX<101.
2	51-55	I5	MNMAX	7	The number of Fourier series coefficients used to describe the initial conditions, pressure and thermal loads. Not applicable in a vibration analysis.
2	56-60	I5	MAXM	8	The number of Fourier series coefficients used in describing the solution. MAXM>=MNMAX, MAXM<11 and MAXM * KMAX <201. Not applicable in a vibration analysis.
2	61-65	I5	LSMAX	1	For a linear analysis.
				99	Use many load steps for a nonlinear static analysis.
				750	For a dynamic analysis LSMAX is the number of time increments, where LSMAX=$T_{max}/\Delta T$. Not applicalbe in a vibration analysis.
2	66-70	I5	LCHMAX	3	The number of load step size reductions in a static analysis. Recommended range is 2-4.

Card	Columns	Format	Item	Example	Meaning
				0	For a dynamic analysis. Not applicable in a vibration analysis
2	71-75	I5	ITRMAX	1	For a linear analysis.
				30	Maximum number of iterations at a load or time step or for a frequency. Up to 30 for a nonlinear analysis and 99 for a vibration analysis. Adjust as desired.
2	76-80	I5	IC	0	Initial conditions. Set to 0 for a static analysis, or for a dynamic analysis for a shell at rest at T = 0.
				1	For a dynamic analysis with initial conditions.
3	1-12	E12.3	NU	0.3	Poisson's ratio, ν.
3	13-24	E12.3	SIGO	1000.0	Reference stress level, σ_o
				1.0	If the input data is dimensional.
3	25-36	E12.3	ELAST	.3E8	Reference modulus of elasticity, E_o.
				1.0	If the input data is dimensinnal.
3	37-48	E12.3	TKN	.4E-2	Reference thickness, h_o.
				1.0	If the input data is dimensional.
3	49-60	E12.3	CHAR	8.6	Characteristic shell dimension, a.
				1.0	If the input data is dimensional.

Card	Columns	Format	Item	Example	Meaning
3	61-72	E12.3	TEEO	0.0	For a static analysis.
				.966E-5	Reference time, T_o for a dynamic analysis.
				10.0	Nondimensionalizing time in a vibration analysis. Select in accordance with Eq (2).
4	1-12	E12.3	DELOAD	0.2	For a static analysis DELOAD is the load increment. It remains unchanged until the solution fails to converge in ITRMAX iterations. It is then reduced by a factor of five. A maximum of LCHMAX reductions will occur before the program terminates.
				.8E-7	For a dynamic analysis DELOAD is the nondimensional time increment. LSMAX such time increments are made. Not applicable in a vibration analysis.
4	13-24	E12.3	EPS	.01	Convergence criterion. Recommended range is .01 to .0001 for most applications. For a vibration analysis a somewhat tighter criterion may be desired.

Include as many cards 5 as necessary to specify the NTHMAX meridians. If NTHMAX = 0 then card 5 is OMITTED.

Card	Columns	Format	Item	Example	Meaning
5	1-72	6E12.3		10.0	A list of circumferential coordinates, θ, in degrees and tenths, at which the solution will be printed. There must be at least NTHMAX entries. Not applicable in a vibration analysis.

Cards 6 through 23 describe the boundary conditions at the first and last stations. The boundary conditions exist on the total variables, not on the individual harmonics. If IBCINL = -1 (an initial pole) then OMIT cards 6 through 14, inclusive. If IBCFNL = -1 (a final pole) then OMIT cards 15 through 23, inclusive. Loadings specified through the edge boundary conditions are taken in the zeroth harmonic only. The column matrix $\{\ell\}$ is set to zero for all other harmonics. The boundary conditions are dimensional. The format of cards 6 through 23 is 4E16.8.

Card 6,15 Card 7,16 Card 8,17 Card 9,18

$$\begin{bmatrix} \Omega(1,1) & \Omega(1,2) & \Omega(1,3) & \Omega(1,4) \\ \Omega(2,1) & \Omega(2,2) & \Omega(2,3) & \Omega(2,4) \\ \Omega(3,1) & \Omega(3,2) & \Omega(3,3) & \Omega(3,4) \\ \Omega(4,1) & \Omega(4,2) & \Omega(4,3) & \Omega(4,4) \end{bmatrix} \begin{Bmatrix} N_s \\ \hat{N}_{s_\theta} \\ \hat{Q}_s \\ \Phi_s \end{Bmatrix} +$$

Card 10,19 Card 11,20 Card 12,21 Card 13,22 Card 14,23

$$\begin{bmatrix} \Lambda(1,1) & \Lambda(1,2) & \Lambda(1,3) & \Lambda(1,4) \\ \Lambda(2,1) & \Lambda(2,2) & \Lambda(2,3) & \Lambda(2,4) \\ \Lambda(3,1) & \Lambda(3,2) & \Lambda(3,3) & \Lambda(3,4) \\ \Lambda(4,1) & \Lambda(4,2) & \Lambda(4,3) & \Lambda(4,4) \end{bmatrix} \begin{Bmatrix} U \\ V \\ W \\ M_s \end{Bmatrix} = \begin{Bmatrix} \ell(1) \\ \ell(2) \\ \ell(3) \\ \ell(4) \end{Bmatrix}$$

Card	Column	Format	Item	Example	Meaning
24	1- 2	L2	$PLOTS	F	Indicates plots are NOT desired.
				T	Indicates plots are desired. KMAX<101
24	3- 4	L2	$MODAL	F	Indicates plots are for summed solutions only
				T	Indicates plots are for modal solutions only.

For the remainder of card 24 entries, 0 indicates that no plots are desired for the particular item, and 1 indicates that they are desired. All graphs are plotted as the indicated item versus the station number. If a complete plot is desired, insure that IFREQ = 1.

24	5- 6	I2	IRADII	1	Plot the radii as computed by subroutine GEOM.
24	7- 8	I2	IGAMMA	1	Plot ρ'/ρ as computed by subroutine GEOM.
24	9-10	I2	IOMEGS	1	Plot ω_s as computed by subroutine GEOM.
24	11-12	I2	IOMEGT	1	Plot ω_θ as computed by subroutine GEOM.
24	13-14	I2	IDEOMS	1	Plot ω_s' ascomputed by subroutine GEOM.
24	15-16	I2	IBSTIF	1	Plot the stiffness d as computed by subroutine BDB.
24	17-18	I2	IDSTIF	1	Plot the stiffness d as computed by the subroutine BDB.
24	19-20	I2	IBBSTF	1	Plot the stiffness $db/d\zeta$ as computed by subroutine BDB.
24	21-22	I2	IDDSTF	1	Plot the stiffness $dd/d\zeta$ as computed by subroutine BDB.
24	23-24	I2	IPR	1	Plot the normal component of the pressure load.

Card	Column	Format	Item	Example	Meaning
24	25-26	I2	IPS	1	Plot the meridional component of the pressure load.
24	27-28	I2	IPT	1	Plot the circumferential component of the pressure load.
24	29-30	I2	ITT	1	Plot the thermal load.
24	31-32	I2	IMT	1	Plot the thermal moment.
24	33-34	I2	IDTT	1	Plot d/dζ of the thermal load.
24	35-36	I2	IDMT	1	Plot d/dζ of the thermal moment.
24	37-38	I2	INS	1	Plot the meridional membrane force distribution.
24	39-40	I2	INTH	1	Plot the circumferential membrane force distribution
24	41-42	I2	INSTH	1	Plot the meridio-circumferential membrane force distribution.
24	43-44	I2	IQS	1	Plot the transverse force distribution.
24	45-46	I2	IMS	1	Plot the meridional moment distribution.
24	47-48	I2	IMTH	1	Plot the circumferential moment distribution.
24	49-50	I2	IMSTH	1	Plot the meridio-circumferential moment distribution.
24	51-52	I2	IU	1	Plot the meridional displacement distribution.
24	53-54	I2	IV	1	Plot the circumferential displacement distribution.

Card	Column	Format	Item	Example	Meaning
24	55-56	I2	IW	1	Plot the normal displacement distribution.
24	57-58	I2	IPHIS	1	Plot the meridional rotation distribution.
24	59-60	I2	IPHIT	1	Plot the circumferential rotation distribution.
24	61-62	I2	IPHI	1	Plot the meridio-circumferential rotation distribution.

- -

Cards 24A and 24B are only included in a SATANS-III vibration analysis and then only when $VIBES is TRUE. If $VIBES is FALSE (no vibration analysis to be performed) then cards 24A and 24B are OMITTED.

24A	1- 5	I5	IVIBES	4	The number of vibration analyses = the number of cards 24B to be included.
24A	6- 9	4F1.0	DIAG	1110	The elements of the diagonal of the mass matrix mu-bar. The 4th element is subsequently automatically set to 0.
24A	10-25	F16.8	EPSVEC	1.E-4	Vector test convergence criterion. If EPSVEC is left = 0.0 it is subsequently automatically set to EPS (card 4).
24A	26-30	I5	ITR2	50	ITR2 is a trouble-shooting aid. If ITR2 is entered every ITR2 iteration the Rayleigh quotient eigenvalue will be printed. If ITR2 is left = 0 it is subsequently automatically set to print the next to last solution in the event of non-convergence.

- -

Card	Column	Format	Item	Example	Meaning

Include one card 24B for each vibration analysis desired. The analyses are done one harmonic at a time and all input information is stated on one card per harmonic. The total number of cards 24B = IVIBES. The information is stored in two arrays of three elements each, MODVIB(J,K) and SHFTPT(J,K) where J=1(1)3 for the Kth analysis and K=1(1)IVIBES.

24B	1- 5	I5	KEY1	0	The lowest frequency in the circumferential Fourier harmonic mode specified by KEY2 will be determined. The vector convergence test will automatically be performed.
				1,2, or 3	1,2 or 3 shift points are supplied by the user. Vector convergence test not performed unless so specified.
				4	A frequency range search will be performed for the mode specified by KEY2.
24B	6-10	I5	KEY2	8	The Fourier harmonic in the circumferential direction for which the vibration analysis will be performed.
24B	11-15	I5	KEY3	0	The vector convergence test will not be performed if KEY1=1,2,3 or 4.
				1	The vector convergence test will be performed.
24B	16-30	F15.8		1.E2	If KEY1=1,2 or 3 this is the first shift point. If KEY1=4 this is the lower bound of the frequency range to be searched. Units are Hz (CPS).

Card	Column	Format	Item	Example	Meaning
24B	31-45	F15.8		2.E4	If KEY1=2 or 3 this is the second shift point. If KEY1=4 this is the upper bound of the frequency range search. Units are Hz (CPS).
24B	46-60	F15.8		4.E3	Applicable only if KEY1=3 in which case this is the third shift point. Units are Hz(CPS).
25	1- 2	I2	IRNAGN	0	Indicates this is the only run by the program.
				1	Indicates another run by the program is to be made. Another set of input data cards must be supplied. The user supplied sub-routines remain the same. The new set of input cards is placed after this card.

APPENDIX F

INPUT AND OUTPUT

PART ONE

USER SUPPLIED PROGRAMS

To use SATANS-III the user must supply 6 programs. The first is a simple 6 card main program that passes control to SATANS.

The second, subroutine GEOM, computes the nondimensional geometry functions of the shell. The correspondence between the nondimensional geometry functions and FORTRAN variables is as follows:

DEL = Δ = $S_T/(a(KMAX-1))$

R(K) = ρ_K = $(r/a)_K$

GAM(K) = γ_K = $(\rho'/\rho)_K$ = $((dr/ds)/(r/a))_K$

OMT(K) = $(\omega_\theta)_K$ = $(a/R_\theta)_K$

OMXI(K) = $(\omega_s)_K$ = $(a/R_s)_K$

DEOMX(K) = $(\omega_s')_K$ = $(a^2 d(R_s^{-1})/ds)_K$

MASS(K) = μ_K

for K = 1(1)KMAX

The third, subroutine BDB, computes the nondimensional stiffness quantities b, b', d and d'. Correspondence between the stiffness quantities and FORTRAN variables is as follows:

$$B = b_k = \int E_y d\zeta / ((1-\nu^2)E_o h_o)$$

$$DB = b'_K = a(dB/ds)_K$$

$$D = d_K = \int \zeta^2 E_y d\zeta / ((1-\nu^2)E_o h_o^3)$$

$$DD = d_K' = a(dD/ds)_K$$

The last three subroutines, PLOAD, TLOAD and INITL compute the nondimensional loads applied to the shell, thermal loads applied to the shell and, in the case of a dynamic analysis, the initial conditions on z and $\frac{\partial z}{\partial t}$. Details on these subroutines are given in Refs. 1 and 2.

APPENDIX G contains example input routines for a vibration analysis.

APPENDIX F

PART TWO

SAMPLE OUTPUT

--PROBLEM NUMBER 7--

CIRCULAR CYLINDRICAL SHELL WITH A LINEARLY VARYING THICKNESS

--INPUT DATA RECORD--

THE BOUNDARY CONDITIONS ARE:

AT THE INITIAL EDGE

OMEGA BAR					LAMDA BAR					EL
(0.100E 01	0.0	0.0	0.0)	NS	(0.0	0.0	0.0	0.0)	U	0.0
(0.0	0.0	0.0	0.0)	NST +	(0.0	0.100E 01	0.0	0.0)	V =	0.0
(0.0	0.0	0.0	0.0)	QS	(0.0	0.0	0.100E 01	0.0)	W	0.0
(0.0	0.0	0.0	0.0)	PHIS	(0.0	0.0	0.0	0.100E 01)	MS	0.0

AT THE FINAL EDGE

OMEGA BAR					LAMDA BAR					EL
(0.100E 01	0.0	0.0	0.0)	NS	(0.0	0.0	0.0	0.0)	U	0.0
(0.0	0.0	0.0	0.0)	NST +	(0.0	0.100E 01	0.0	0.0)	V =	0.0
(0.0	0.0	0.0	0.0)	QS	(0.0	0.0	0.100E 01	0.0)	W	0.0
(0.0	0.0	0.0	0.0)	PHIS	(0.0	0.0	0.0	0.100E 01)	MS	0.0

NUMBER OF STATIONS---------- 27
MAXIMUM NUMBER OF ITERATIONS---------- 100
NUMBER OF VIBRATION ANALYSES---------- 1
CONVERGENCE CRITERION (EIGENVALUES)----- 0.1000E-05

CHARACTERISTIC SHELL DIMENSION---------- 0.1000E 01
REFERENCE THICKNESS---------- 0.1000E 01
REFERENCE ELASTICITY---------- 0.1000E 01
REFERENCE STRESS---------- 0.1000E 01
REFERENCE TIME---------- 0.1000E 02
POISSON'S RATIO---------- 0.3000E 00

PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED GEOMETRY, STIFFNESS OR LOADING QUANTITIES:
D-STIFFNESS

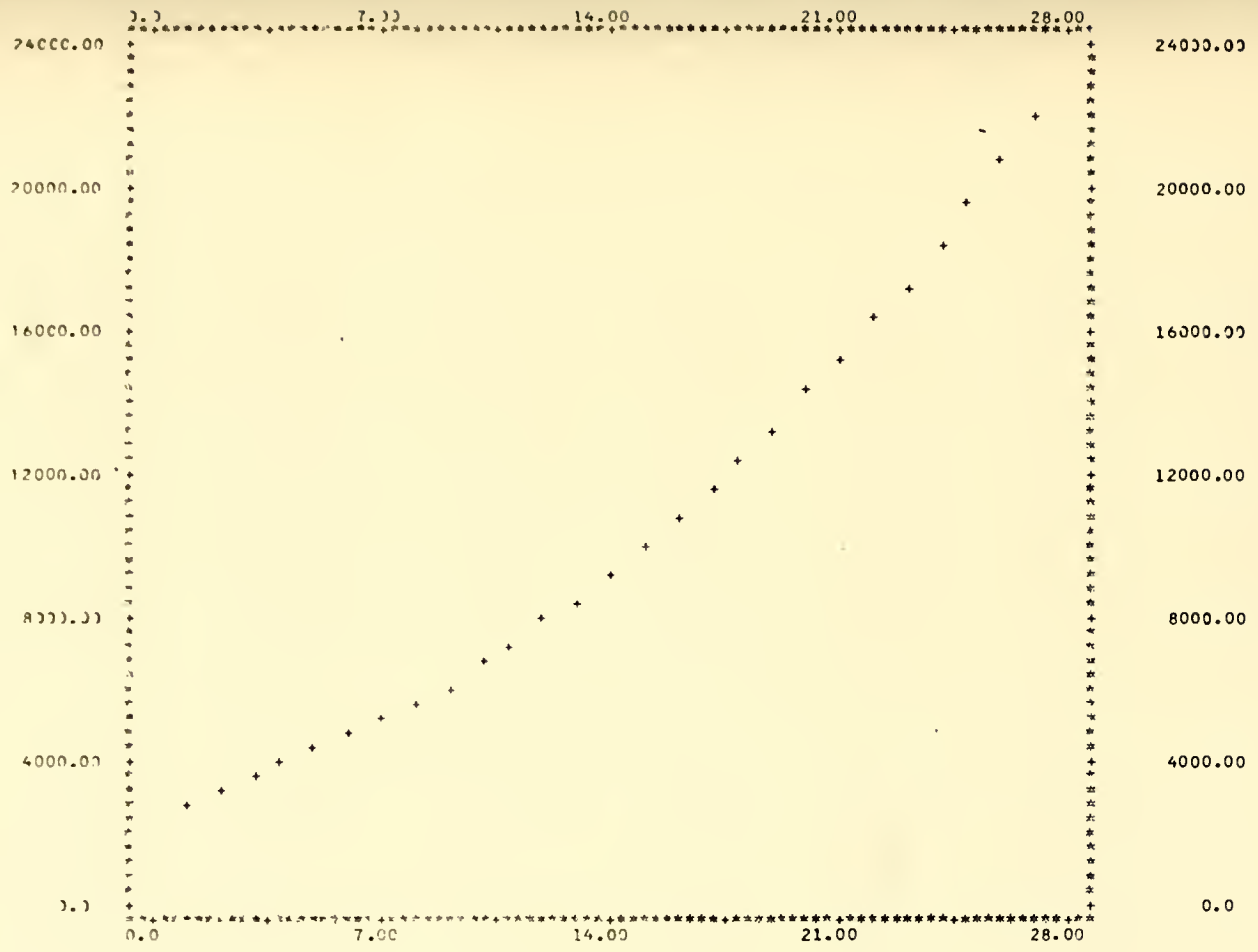
PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED MODAL QUANTITIES:
W

THE DATA PRINTED IS DIMENSIONAL

EXECUTING IN SUBROUTINE "DYNAMIC"

STATION	RADIUS	GAMMA	OMEGA S	OMEGA THETA	DEOMEGA S	MASS
1	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.7330E-04
2	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.7612E-04
3	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.7894E-04
4	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.8176E-04
5	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.8458E-04
6	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.8740E-04
7	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.9022E-04
8	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.9303E-04
9	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.9585E-04
10	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.9867E-04
11	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1015E-03
12	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1043E-03
13	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1071E-03
14	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1099E-03
15	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1128E-03
16	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1156E-03
17	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1184E-03
18	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1212E-03
19	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1240E-03
20	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1269E-03
21	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1297E-03
22	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1325E-03
23	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1353E-03
24	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1381E-03
25	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1410E-03
26	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1438E-03
27	0.1000E 03	0.0	0.0	0.1000E-01	0.0	0.1466E-03

STATION	B STIFFNESS	D STIFFNESS	B PRIME	D PRIME
1	0.329670E 07	0.274725E 04	0.109890E 05	0.274725E 02
2	0.342350E 07	0.307658E 04	0.109890E 05	0.296263E 02
3	0.355029E 07	0.343124E 04	0.109890E 05	0.318616E 02
4	0.367709E 07	0.381217E 04	0.109890E 05	0.341780E 02
5	0.380388E 07	0.422028E 04	0.109890E 05	0.365757E 02
6	0.393068E 07	0.465653E 04	0.109890E 05	0.390548E 02
7	0.405748E 07	0.512186E 04	0.109890E 05	0.416151E 02
8	0.418427E 07	0.561719E 04	0.109890E 05	0.442567E 02
9	0.431107E 07	0.614347E 04	0.109890E 05	0.469795E 02
10	0.443787E 07	0.670164E 04	0.109890E 05	0.497837E 02
11	0.456466E 07	0.729264E 04	0.109890E 05	0.526691E 02
12	0.469146E 07	0.791740E 04	0.109890E 05	0.556358E 02
13	0.481825E 07	0.857686E 04	0.109890E 05	0.586838E 02
14	0.494505E 07	0.927196E 04	0.109890E 05	0.618131E 02
15	0.507185E 07	0.100036E 05	0.109890E 05	0.650236E 02
16	0.519864E 07	0.107728E 05	0.109890E 05	0.683154E 02
17	0.532544E 07	0.115804E 05	0.109890E 05	0.716885E 02
18	0.545224E 07	0.124275E 05	0.109890E 05	0.751429E 02
19	0.557903E 07	0.133148E 05	0.109890E 05	0.786785E 02
20	0.570583E 07	0.142434E 05	0.109890E 05	0.822954E 02
21	0.583262E 07	0.152143E 05	0.109890E 05	0.859937E 02
22	0.595942E 07	0.162282E 05	0.109890E 05	0.897732E 02
23	0.608622E 07	0.172863E 05	0.109890E 05	0.936340E 02
24	0.621301E 07	0.183893E 05	0.109890E 05	0.975760E 02
25	0.633981E 07	0.195383E 05	0.109890E 05	0.101599E 03
26	0.646660E 07	0.207342E 05	0.109890E 05	0.105704E 03
27	0.659340E 07	0.219779E 05	0.109890E 05	0.109890E 03



X-SCALE: "*"= 0.350E 00 UNITS

Y-SCALE: "*"= 0.400E 03 UNITS

E-STIFFNESS VS STATION

EXECUTING IN SUBROUTINE VIBERS

PERFORMING FREE VIBRATION ANALYSES

THE LOWEST FREQUENCY IN HARMONIC 9 IS DETERMINED

DIAGONAL OF MASS MATRIX (MU-BAR) IS (1. 1. 1. 0.)

THE CONVERGENCE TEST ON THE VECTOR WILL BE PERFORMED
CONVERGENCE CRITERION IS 0.100-05

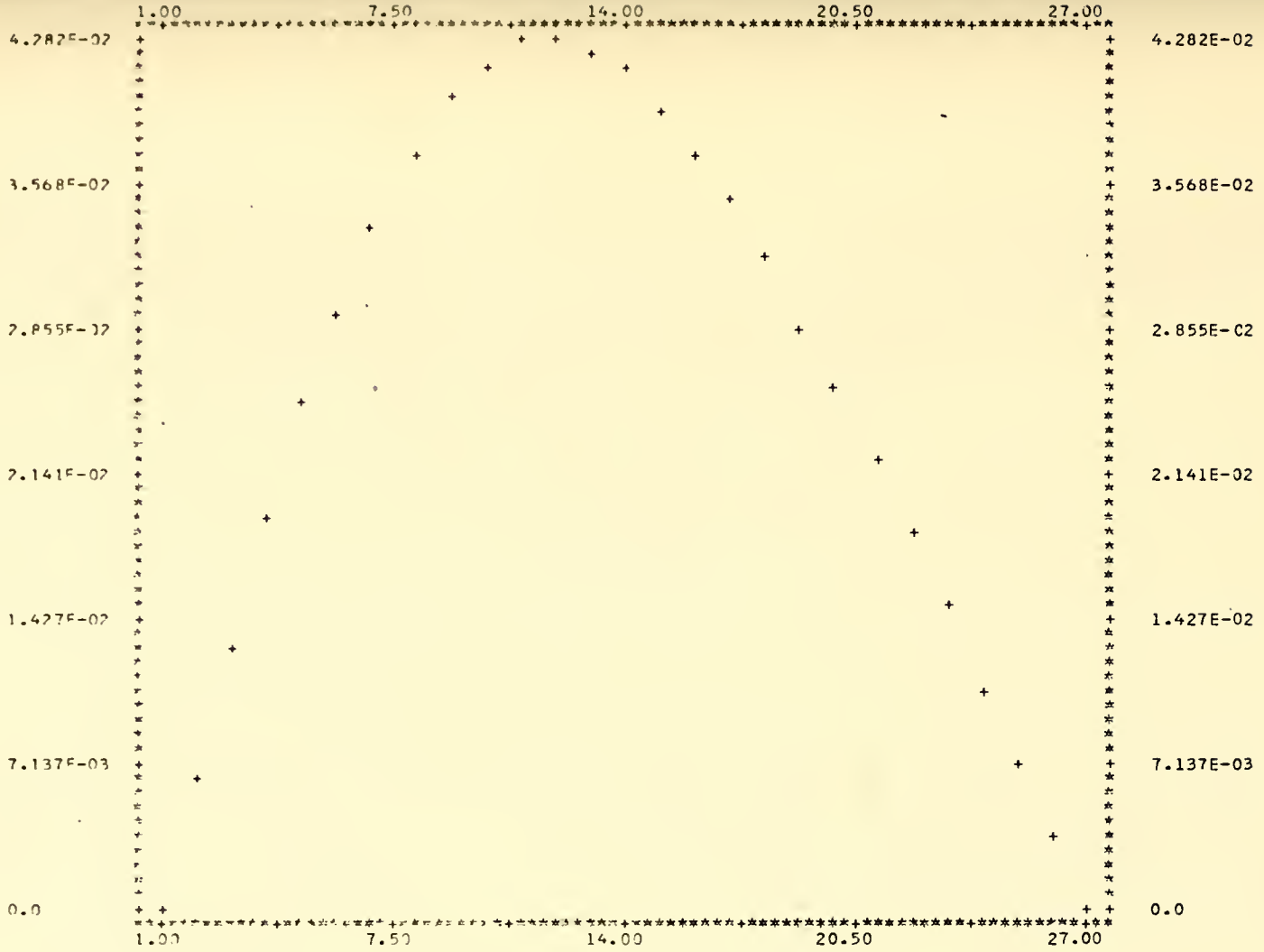
SHIFT POINT IS 0.0
(0.) HZ)

CONVERGED SOLUTION. ITERATIONS = 19

RAYLEIGH QUOTIENT FREQUENCY = 0.1259931572790I0 02 HZ

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	0.1663E-03	0.4989E-04	-0.4966E 01	0.1450E-01	0.0	0.0	0.9814E-01
2	0.5065E 01	0.2035E 00	-0.4808E 01	0.1481E-01	0.5361E-01	0.1644E 00	0.1082E 00
3	0.9936E 01	0.3670E 00	-0.4562E 01	0.1627E-01	0.1169E 00	0.3609E 00	0.1154E 00
4	0.1447E 02	0.5116E 00	-0.4161E 01	0.1733E-01	0.1893E 00	0.5867E 00	0.1191E 00
5	0.1853E 02	0.6273E 00	-0.3627E 01	0.1789E-01	0.2694E 00	0.8376E 00	0.1189E 00
6	0.2197E 02	0.7124E 00	-0.2986E 01	0.1788E-01	0.3552E 00	0.1108E 01	0.1144E 00
7	0.2471E 02	0.7635E 00	-0.2270E 01	0.1726E-01	0.4445E 00	0.1391E 01	0.1052E 00
8	0.2668E 02	0.7800E 00	-0.1512E 01	0.1600E-01	0.5350E 00	0.1680E 01	0.9141E-01
9	0.2787E 02	0.7659E 00	-0.7427E 00	0.1411E-01	0.6240E 00	0.1966E 01	0.7312E-01
10	0.2826E 02	0.7148E 00	0.4214E-02	0.1161E-01	0.7088E 00	0.2241E 01	0.5064E-01
11	0.2790E 02	0.6377E 00	0.6971E 00	0.8568E-02	0.7869E 00	0.2495E 01	0.2446E-01
12	0.2687E 02	0.5434E 00	0.1319E 01	0.5041E-02	0.8557E 00	0.2725E 01	-0.4819E-02
13	0.2524E 02	0.4191E 00	0.1933E 01	0.1112E-02	0.9132E 00	0.2918E 01	-0.3648E-01
14	0.2313E 02	0.2904E 00	0.2238E 01	-0.3091E-02	0.9573E 00	0.3071E 01	-0.6974E-01
15	0.2066E 02	0.1635E 00	0.2522E 01	-0.7497E-02	0.9867E 00	0.3177E 01	-0.1038E 00
16	0.1796E 02	0.2480E-01	0.2682E 01	-0.1200E-01	0.1000E 01	0.3233E 01	-0.1380E 00
17	0.1515E 02	-0.1028E 00	0.2719E 01	-0.1650E-01	0.9964E 00	0.3234E 01	-0.1715E 00
18	0.1236E 02	-0.2114E 00	0.2641E 01	-0.2092E-01	0.9756E 00	0.3179E 01	-0.2039E 00
19	0.9709E 01	-0.3098E 00	0.2463E 01	-0.2521E-01	0.9373E 00	0.3066E 01	-0.2347E 00
20	0.7284E 01	-0.3874E 00	0.2198E 01	-0.2933E-01	0.8814E 00	0.2894E 01	-0.2637E 00
21	0.5165E 01	-0.4344E 00	0.1869E 01	-0.3326E-01	0.8082E 00	0.2664E 01	-0.2908E 00
22	0.3410E 01	-0.4564E 00	0.1497E 01	-0.3703E-01	0.7177E 00	0.2374E 01	-0.3161E 00
23	0.2047E 01	-0.4579E 00	0.1112E 01	-0.4069E-01	0.6101E 00	0.2025E 01	-0.3401E 00
24	0.1177E 01	-0.4368E 00	0.7423E 00	-0.4428E-01	0.4852E 00	0.1615E 01	-0.3631E 00
25	0.4634E 00	-0.3888E 00	0.4191E 00	-0.4792E-01	0.3424E 00	0.1143E 01	-0.3859E 00
26	0.1380E 00	-0.2122E 00	0.1774E 00	-0.5163E-01	0.1808E 00	0.6057E 00	-0.4089E 00
27	-0.2915E-04	0.1046E-03	0.8086E-01	-0.5238E-01	0.4938E-06	0.3568E-06	-0.4330E 00

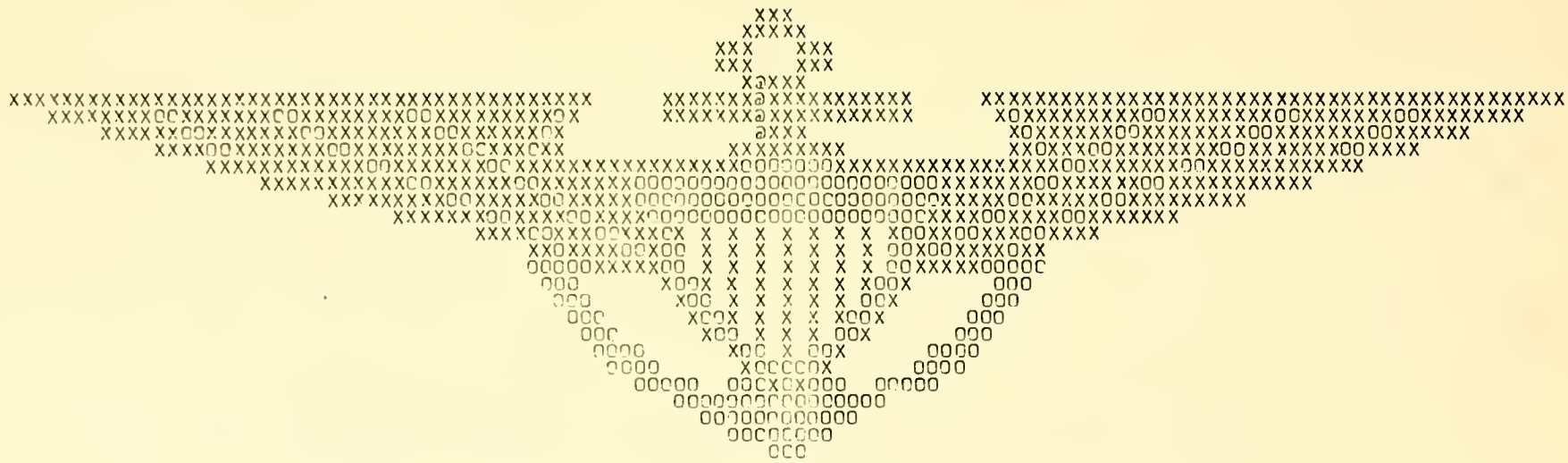
STATION	U	V	W	PHI S	PHI THETA	PHI
1	-0.6673E-03	0.0	0.0	-0.5741E-03	0.0	-0.6216E-04
2	-0.6576E-03	-0.7407E-03	0.6624E-02	-0.5652E-03	0.5887E-03	-0.6119E-04
3	-0.6302E-03	-0.1458E-02	0.1304E-01	-0.5404E-03	0.1159E-02	-0.5856E-04
4	-0.5874E-03	-0.2134E-02	0.1900E-01	-0.5022E-03	0.1697E-02	-0.5448E-04
5	-0.5315E-03	-0.2753E-02	0.2463E-01	-0.4528E-03	0.2189E-02	-0.4919E-04
6	-0.4651E-03	-0.3301E-02	0.2954E-01	-0.3947E-03	0.2626E-02	-0.4295E-04
7	-0.3911E-03	-0.3769E-02	0.3374E-01	-0.3301E-03	0.2999E-02	-0.3600E-04
8	-0.3121E-03	-0.4150E-02	0.3716E-01	-0.2615E-03	0.3303E-02	-0.2851E-04
9	-0.2308E-03	-0.4441E-02	0.3978E-01	-0.1913E-03	0.3535E-02	-0.2102E-04
10	-0.1495E-03	-0.4641E-02	0.4158E-01	-0.1214E-03	0.3695E-02	-0.1346E-04
11	-0.7054E-04	-0.4752E-02	0.4258E-01	-0.5390E-04	0.3785E-02	-0.6130E-05
12	0.4409E-05	-0.4777E-02	0.4282E-01	0.9772E-05	0.3806E-02	0.7967E-06
13	0.7381E-04	-0.4724E-02	0.4235E-01	0.6834E-04	0.3765E-02	0.7187E-05
14	0.1366E-03	-0.4599E-02	0.4124E-01	0.1209E-03	0.3666E-02	0.1294E-04
15	0.1992E-03	-0.4410E-02	0.3956E-01	0.1668E-03	0.3517E-02	0.1799E-04
16	0.2596E-03	-0.4167E-02	0.3739E-01	0.2059E-03	0.3324E-02	0.2230E-04
17	0.2795E-03	-0.3879E-02	0.3481E-01	0.2381E-03	0.3094E-02	0.2589E-04
18	0.3119E-03	-0.3553E-02	0.3190E-01	0.2627E-03	0.2835E-02	0.2876E-04
19	0.3373E-03	-0.3195E-02	0.2873E-01	0.2833E-03	0.2553E-02	0.3099E-04
20	0.3565E-03	-0.2824E-02	0.2538E-01	0.2975E-03	0.2254E-02	0.3263E-04
21	0.3702E-03	-0.2432E-02	0.2186E-01	0.3071E-03	0.1943E-02	0.3378E-04
22	0.3795E-03	-0.2033E-02	0.1827E-01	0.3131E-03	0.1624E-02	0.3451E-04
23	0.3853E-03	-0.1628E-02	0.1464E-01	0.3162E-03	0.1301E-02	0.3494E-04
24	0.3886E-03	-0.1221E-02	0.1098E-01	0.3173E-03	0.9758E-03	0.3514E-04
25	0.3902E-03	-0.8135E-03	0.7314E-02	0.3174E-03	0.6501E-03	0.3521E-04
26	0.3908E-03	-0.4064E-03	0.3654E-02	0.3169E-03	0.3248E-03	0.3521E-04
27	0.3910E-03	0.6767E-10	0.1280E-08	0.3166E-03	0.1159E-09	0.3521E-04

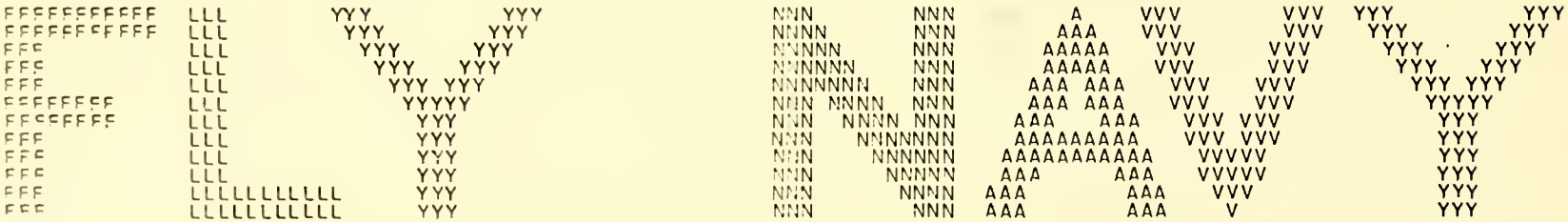


X-SCALE: "*"= 0.325E 00 UNITS

Y-SCALE: "*"= 0.714E-03 UNITS

NORMAL DEFLECTION VS STATION





APPENDIX G

EXAMPLES OF USER SUPPLIED SUBROUTINES

AND

A LISTING OF SATANS-III


```
C**********************************************************************
C     THIS PROGRAM SERVES AS THE 'MAIN' PROGRAM, AND CALLS 'SATANS',   *
C     AND CHECKS 'IRNAGN' TO SEE IF WE DESIRE ANOTHER RUN              *
C**********************************************************************
      COMMON /BLRUN/ IRNAGN
C**********************************************************************
    1 CALL SATANS
      IF (IRNAGN.EQ.1) GO TO 1
      CALL FLYNVY
      STOP
      END
```

```
      SUBROUTINE GEOM
C*****************************************************************
C     THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL                *
C     GEOMETRY FUNCTIONS OF THE SHELL.                           *
C*****************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL NU,LAM,LAM2,JAY,MT,LSD1S,LSD1N,MASS
      COMMON /IBL2/NN(10),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /BL8/  R(200),GAM(200),OMT(200)
      COMMON /BL11/ OMXI(200),PHEE,T0,T2
      COMMON /BL17/ DEL
      COMMON /BL20/ DEOMX(200)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL100/ TEE0,SOYNMC
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(200)
C*****************************************************************
C     CIRCULAR CYLINDRICAL SHELL WITH A LINEARLY VARYING THICKNESS  *
C     LENGTH (ST)     = 300.0      INCHES                           *
C     RADIUS (RAD)    = 100.0      INCHES                           *
C     THICKNESS   H1  = 0.1        INCHES                           *
C                 H2  = 0.1 * X    INCHES                           *
C     DENSITY(DNSTY)  = .733E-03   LB-SEC**2/INCHES**4              *
C*****************************************************************
      X             = 2.
      DNSTY         = .733E-03
      H1            = .1
      H2            = .1 * X
      RAD           = 100.
      ST            = 300.
      DEL           = ST / FLOAT(KMAX - 1)
      SLOPE         = (H2-H1)/ST
      DO  1  K      = 1,KMAX
      R(K)          = RAD/CHAR
      GAM(K)        = 0.0
      OMT(K)        = CHAR/RAD
      OMXI(K)       = 0.0
      DEOMX(K)      = 0.0
      S             = DEL * FLOAT(K-1)
      H             = S * SLOPE + H1
      MASS(K)       = (DNSTY*H*CHAR**2)/(TKN*ELAST*TEE0**2)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BDB(K,B,DB,D,DD)
C*****************************************************************
C     THIS SUBROUTINE COMPUTES THE NONDIMENSIONAL                *
C     IN-PLANE AND BENDING STIFFNESS OF THE SHELL.               *
C*****************************************************************
      REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),
     1              V3(10),W3(10)
      COMMON /BL17/ DEL
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
C*****************************************************************
C     B(K) =  (E * H)/((1.-NU**2)*(E(0)*H(0)))                   *
C     D(K) = (E * H**3)/(12.*(1.-NU**2)*E(0)*H(0)**3)            *
C*****************************************************************
      NU          = .3
      E           = .358
      X           = 2.
      S           = DEL * FLOAT(K-1)
      SLOPE       = .1*(X-1.)/300.
      H           = S*SLOPE + .1
      B           = (E*H)/((1.-NU**2)*ELAST*TKN)
      DB          = (E*SLOPE*CHAR)/((1.-NU**2)*ELAST*TKN)
      D           = (E*H**3)/((1.-NU**2)*12.*ELAST*TKN**3)
      DD          = (E*(H**2)*SLOPE*CHAR)/((1.-NU**2)*4.*ELAST*TKN**3)
      RETURN
      END


      SUBROUTINE PLOAD(K)
C*****************************************************************
C         THIS SUBROUTINE ESTABLISHES THE NON-DIMENSIONAL FOURIER *
C         COEFFICIENTS OF THE LOADS APPLIED TO THE SHELL          *
C*****************************************************************
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/NN(10),MNIMIT
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL8/ LSTEP,ITR
      COMMON /BL3/  PR(10),PX(10),PT(10)
      COMMON /BL6/ SOE,OSE,ALOAD
      COMMON /BL8/  R(200),GAM(200),OMT(200)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL102/ DELOAD
      COMMON /BL103/ MASS(20)
C*****************************************************************
      RETURN
      END
```

```
      SUBROUTINE TLOAD(K)
C*****************************************************************
C     THIS SUBROUTINE DESCRIBES THE THERMAL LOADING ON THE SHELL      *
C*****************************************************************
      REAL NU
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ NN(10),MNINIT
      COMMON /IBL8/ LSTEP,ITR
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)
      COMMON /BL6/ SOE,OSE,ALOAD
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),
     1              V3(10),W3(10)
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
C*****************************************************************
      RETURN
      END

      SUBROUTINE INITL
C*****************************************************************
C     THIS SUBROUTINE DESCRIBES THE INITIAL CONDITIONS FOR DYNAMIC CASES
C*****************************************************************
      IMPLICIT LOGICAL*1 ($)
      COMMON /IBL1/ MNMAX
      COMMON /IBL2/ NN(10),MNINIT
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL9/ MAXM
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /BL6/ SOE,OSE,ALOAD
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO
      COMMON /BL100/ TEEC,$DYNMC
      COMMON /BL101/ DELSD
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),Z0(4,220),Z2(4,220),
     1               Z3(4,220)
C*****************************************************************
      RETURN
      END
```

```
*****************************************************************************
*                  THE FOLLOWING SET OF CARDS IS AN EXAMPLE                 *
*                  OF INPUT CARDS FOR A VIBRATION ANALYSIS.                 *
*****************************************************************************
    5   10   15   20   25   30   35   40   45   50   55   60   65   70   75   80
*****************************************************************************


CIRCULAR CYLINDRICAL SHELL WITH A LINEARLY VARYING THICKNESS
    7  T T    1    0    0    1    1    0    0   27    1    1    1    0  100    0
     .3         1.          1.          1.          1.          10.
   1.0       0.000001
     1.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            1.0            0.0            0.0
     0.0            0.0            1.0            0.0
     0.0            0.0            0.0            1.0
     0.0            0.0            0.0            0.0
     1.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            0.0            0.0            0.0
     0.0            1.0            0.0            0.0
     0.0            0.0            1.0            0.0
     0.0            0.0            0.0            1.0
     0.0            0.0            0.0            0.0
 T T            1                                  1
    11110
        9
 0
```

SATANS-III LISTING

THE SUBROUTINES ARE LISTED, READING ACROSS, AS FOLLOWS:

NAME	CARD NUMBER	NAME	CARD NUMBER	NAME	CARD NUMBER	NAME	CARD NUMBER
SATANS	10	STATIC	2540	DYNMAC	5610	VIBERS	8680
PMATRX	9240	XANDZ	11560	FORCE	14270	HJ	15890
FFG	16610	ABC	17670	PANDD	18000	MODES	18370
POLE	19360	INLPOL	20880	FNLPOL	21560	PHIBET	22210
TEAETA	23110	UPDATE	24140	OUTPUT	24520	PLOT1	28340
PLOT2	29410	PLOTIT	31380	SCALIT	31950	ROUND	32740
DRAWIT	33220	RANGER	34720	ADJUST	35540	BTTM	36590
TOPPER	37370	ITREAL	38050	ITRATE	38520	ITCPLX	41130
INITAA	42000	VECOUT	43060	START	44780	MATINV	45150
		FLYNVY	46560				

```
      SUBROUTINE SATANS
C*****************************************************************
C     THIS SUBROUTINE READS ALL DATA, PRINTS THE OUTPUT TITLE PAGE,   *
C     AND PASSES CONTROL TO ONE OF THE MAJOR CONTROLLING SUBROUTINES  *
C*****************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL*8 EPSVIB,EPSVEC,EPSAIT
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
      COMMON /IBL1/ MNMAX
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL5/ IBCINL,IBCFNL
      COMMON /IBL7/ MNMAXC,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)
      COMMON /IBL9/ MAXM
      COMMON /IBL10/ IFREC,NTHMAX
```

```
      COMMON /IBL13/ ITRMAX,LSMAX                                       160
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),        170
     1              UNIT(4,4)                                           180
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10), 190
     1              V3(10),W3(10)                                       200
      COMMON /BL16/ EPS                                                 210
      COMMON /BL18/ EL1(4),ELL(4)                                       220
      COMMON /BL19/ TH(36)                                              230
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                 240
      COMMON /BL100/ TEEO,$DYNMC                                        250
      COMMON /BL102/ DELOAD                                             260
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220), 270
     1              Z3(4,220)                                           280
      COMMON /BLDATA/ TITLE,NL,IMODE,NDIMEN,IPRINT,LCHMAX,IC            290
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF, 300
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,  310
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,     320
     3                IPHIT,IPHI,$PLOTS,$MODAL                          330
      COMMON /BLRUN/ IRNAGN                                             340
      COMMON /BLK1/ VMASS,NN2,$VIBES                                    350
      COMMON /BLK2/ IVIBES,IVS                                          360
      COMMON /BLK8/ EPSVIB,EPSVEC,EPSAIT                                370
      COMMON /BLK9/ ITR1,ITR2                                           380
      COMMON /BLK11/ DIAG(4)                                            390
      DIMENSION MODVIB(3,100),SHFTPT(3,100),TITLE(18)                   400
      EQUIVALENCE (MODVIB(1,1),IS(1,1)),(SHFTPT(1,1),Z3(1,146))         410
C*******************************************************************  420
C     READ IN DATA FOR THIS RUN                                         *430
C*******************************************************************  440
      READ (5,100) TITLE                                                450
      READ(5,101)NO,$DYNMC,$VIBES,IMODE,NDIMEN,NTHMAX,IFREQ,IPRINT,     460
     1IBCINL,IBCFNL,KMAX,MNMAX,MAXM,LSMAX,LCHMAX,ITRMAX,IC              470
      READ (5,102) NU,SIGO,ELAST,TKN,CHAR,TEEO                          480
      READ (5,103) DELOAD,EPS                                           490
      IF (NTHMAX.EQ.0) GO TO 1                                          500
      READ (5,104) (TH(NTH),NTH=1,NTHMAX)                               510
      DO 2 NTH=1,NTHMAX                                                 520
    2 TH(NTH)=TH(NTH)*6.2831853/360.0                                   530
    1 IF (IBCINL.EQ.0) READ (5,105) OMEG1,CAPL1,EL1                     540
      IF (IBCFNL.EQ.0) READ (5,105) OMEGL,CAPLL,ELL                     550
      READ (5,106) $PLOTS,$MODAL,IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,    560
     1             IBSTIF,IDSTIF,IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT, 570
     2             IDMT,INS,INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,   580
     3             IPHIS,IPHIT,IPHI                                   590
      IF(.NOT.$VIBES) GO TO 10                                          600
      READ(5,108) IVIBES,DIAG,EPSVEC,ITR2                               610
      READ(5,109) (((MODVIB(J,I),J=1,3),(SHFTPT(K,I),K=1,3)),I=1,IVIBES) 620
   10 READ (5,107) IRNAGN                                               630
```

```
      IF($VIBES) CALL INITAA
C*****************************************************************
C     THIS CONCLUDES DATA READ-IN FOR THIS RUN.                   *
C     PRINT OUT PERTINENT DATA.                                   *
C*****************************************************************
      WRITE (6,200) NO
      WRITE (6,201) TITLE
      WRITE (6,202)
      IF (IBCINL.LT.0) WRITE (6,203)
      IF (IBCINL.LT.0) GO TO 3
      WRITE (6,204)
      WRITE (6,205) (OMEG1(1,J),J=1,4),(CAPL1(1,J),J=1,4),EL1(1)
      WRITE (6,206) (OMEG1(2,J),J=1,4),(CAPL1(2,J),J=1,4),EL1(2)
      WRITE (6,207) (OMEG1(3,J),J=1,4),(CAPL1(3,J),J=1,4),EL1(3)
      WRITE (6,208) (OMEG1(4,J),J=1,4),(CAPL1(4,J),J=1,4),EL1(4)
    3 IF (IBCFNL.LT.0) WRITE (6,209)
      IF (IBCFNL.LT.0) GO TO 4
      WRITE(6,211)
      WRITE (6,210)
      WRITE (6,205) (OMEGL(1,J),J=1,4),(CAPLL(1,J),J=1,4),ELL(1)
      WRITE (6,206) (OMEGL(2,J),J=1,4),(CAPLL(2,J),J=1,4),ELL(2)
      WRITE (6,207) (OMEGL(3,J),J=1,4),(CAPLL(3,J),J=1,4),ELL(3)
      WRITE (6,208) (OMEGL(4,J),J=1,4),(CAPLL(4,J),J=1,4),ELL(4)
    4 WRITE (6,211)
      IF($VIBES) GO TO 11
      IF ($DYNMC) GO TO 5
      WRITE (6,212) KMAX,MAXM,DELOAD,LSMAX,ITRMAX,LCHMAX,EPS
      WRITE (6,213) CHAR,TKN,ELAST,SIGO,NU
      GO TO 6
    5 WRITE (6,246) KMAX,MAXM,ITRMAX,LSMAX,DELOAD,EPS
      WRITE (6,247) CHAR,TKN,ELAST,SIGO,TEEO,NU
      GO TO 6
   11 WRITE(6,252) KMAX,ITR1,IVIBES,EPS
      WRITE (6,247) CHAR,TKN,ELAST,SIGO,TEEO,NU
    6 IF(.NOT.$PLOTS) GO TO 9
      ICHCK1=IABS(IRADII)+IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)
     1       +IABS(IDEOMS)+IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)
     2       +IABS(IDDSTF)+IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)
     3       +IABS(IMT)+IABS(IDTT)+IABS(IDMT)
      IF (ICHCK1.GT.0) WRITE (6,214)
      IF (ICHCK1.EQ.0) GO TO 8
      IF (IRADII.NE.0) WRITE (6,215)
      IF (IGAMMA.NE.0) WRITE (6,216)
      IF (IOMEGS.NE.0) WRITE (6,217)
      IF (IOMEGT.NE.0) WRITE (6,218)
      IF (IDEOMS.NE.0) WRITE (6,219)
      IF (IBSTIF.NE.0) WRITE (6,220)
      IF (IDSTIF.NE.0) WRITE (6,221)
```

```
      IF (IBBSTF.NE.0) WRITE (6,222)                                    1120
      IF (IDDSTF.NE.0) WRITE (6,223)                                    1130
      IF (IPR   .NE.0) WRITE (6,224)                                    1140
      IF (IPS   .NE.0) WRITE (6,225)                                    1150
      IF (IPT   .NE.0) WRITE (6,226)                                    1160
      IF (ITT   .NE.0) WRITE (6,227)                                    1170
      IF (IMT   .NE.0) WRITE (6,228)                                    1180
      IF (IDTT  .NE.0) WRITE (6,229)                                    1190
      IF (IDMT  .NE.0) WRITE (6,230)                                    1200
    8 ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)       1210
     1       +IABS(IMTH)+IABS(IMSTH)+IABS(IU)+IABS(IV)+IABS(IW)         1220
     2       +IABS(IPHIS)+IABS(IPHIT)+IABS(IPHI)                        1230
      IF((ICHCK1.GT.0).AND.($MODAL) WRITE (6,231)                       1240
      IF((ICHCK1.GT.0).AND..NOT.$MODAL) WRITE (6,232)                   1250
      IF (ICHCK1.EQ.0) GO TO 9                                          1260
      IF (INS   .NE.0) WRITE (6,233)                                    1270
      IF (INTH  .NE.0) WRITE (6,234)                                    1280
      IF (INSTH.NE.0) WRITE (6,235)                                     1290
      IF (IQS   .NE.0) WRITE (6,236)                                    1300
      IF (IMS   .NE.0) WRITE (6,237)                                    1310
      IF (IMTH  .NE.0) WRITE (6,238)                                    1320
      IF (IMSTH.NE.0) WRITE (6,239)                                     1330
      IF (IU    .NE.0) WRITE (6,240)                                    1340
      IF (IV    .NE.0) WRITE (6,241)                                    1350
      IF (IW    .NE.0) WRITE (6,242)                                    1360
      IF (IPHIS.NE.0) WRITE (6,243)                                     1370
      IF (IPHIT.NE.0) WRITE (6,244)                                     1380
      IF (IPHI  .NE.0) WRITE (6,245)                                    1390
    9 IF(NTHMAX.EQ.0) GO TO 7                                           1400
      WRITE (6,248)                                                     1410
      WRITE (6,249) (TH(NTH),NTH=1,NTHMAX)                              1420
    7 IF (NDIMEN.EQ.1) WRITE (6,250)                                    1430
      IF (NDIMEN.EQ.0) WRITE (6,251)                                    1440
C*****************************************************************     1450
C     CALL APPROPRIATE CONTROLLING SUBPROGRAM                       *   1460
C*****************************************************************     1470
      IF ($DYNMC) CALL DYNAMC                                           1480
      IF (.NOT.$DYNMC) CALL STATIC                                      1490
C*****************************************************************     1500
C     READ STATEMENT FORMATS FOLLOW                                 *   1510
C*****************************************************************     1520
  100 FORMAT (18A4)                                                     1530
  101 FORMAT(I5,L3,L2,14I5)                                             1540
  102 FORMAT (6E12.4)                                                   1550
  103 FORMAT (2E12.4)                                                   1560
  104 FORMAT (6F12.4)                                                   1570
  105 FORMAT (4E16.8)                                                   1580
  106 FORMAT (2L2,29I2)                                                 1590
```

```
  107 FORMAT (I2)                                                          1600
  108 FORMAT(I5,4F1.0,F16.8,I5)                                            1610
  109 FORMAT(3I5,3F15.8)                                                   1620
C*****************************************************************       1630
C     WRITE STATEMENT FORMATS FOLLOW                                      *1640
C*****************************************************************       1650
  200 FORMAT ('1',48X,'--PROBLEM NUMBER',I5,'--',///)                      1660
  201 FORMAT (' ',30X,18A4,///)                                            1670
  202 FORMAT (' ',49X,'--INPUT DATA RECORD--',//'     THE BOUNDARY CONDI   1680
     1TIONS ARE: ',//)                                                     1690
  203 FORMAT (' ',T10,'THE SHELL HAS AN INITIAL POLE',/)                   1700
  204 FORMAT (' ',T10,'AT THE INITIAL EDGE',//2X,'----------------OMEGA B   1710
     1AR----------------',15X,'----------------LAMDA BAR----------------',1 1720
     22X,'---EL-----',/)                                                   1730
  205 FORMAT (' ','(',4E10.3,')  NS          (',4E10.3,')  U         ',E10 1740
     1.3)                                                                  1750
  206 FORMAT (' ','(',4E10.3,')  NST   +     (',4E10.3,')  V     =   ',E10 1760
     1.3)                                                                  1770
  207 FORMAT (' ','(',4E10.3,')  QS          (',4E10.3,')  W         ',E10 1780
     1.3)                                                                  1790
  208 FORMAT (' ','(',4E10.3,')  PHIS        (',4E10.3,')  MS        ',E10 1800
     1.3)                                                                  1810
  209 FORMAT (' ',T10,'THE SHELL HAS A FINAL POLE',/)                      1820
  210 FORMAT (' ',T10,'AT THE FINAL EDGE',//2X,'----------------OMEGA BAR   1830
     1----------------',15X,'----------------LAMDA BAR----------------',12 1840
     2X,'---EL-----',/)                                                    1850
  211 FORMAT (////)                                                        1860
  212 FORMAT (' ',4X,'NUMBER OF STATIONS----------------------',I3/5X,'N   1870
     1UMBER OF MODES----------------------',I3/5X,'INCREMENTAL LOAD F      1880
     2ACTOR----------------',F6.3/5X,'MAXIMUM NUMBER OF LOAD STEPS-----   1890
     3-------',I3/5X,'MAXIMUM NUMBER OF ITERATIONS------------',I3/5X,'M   1900
     4AXIMUM NUMBER OF LOAD FACTOR CHANGES---',I3/5X,'CONVERGENCE CRITER   1910
     5ION-------------------',E12.4//)                                     1920
  213 FORMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION----------',E12.4/5X   1930
     1,'REFERENCE THICKNESS----------------------',E12.4/5X,'REFERENCE EL  1940
     2ASTICITY--------------------',E12.4/5X,'REFERENCE STRESS----------   1950
     3--------------',E12.4/5X,'POISSONS RATIO-------------------------    1960
     4',E12.4//)                                                           1970
  214 FORMAT (' ',T10,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED GE   1980
     1OMETRY, STIFFNESS OR LOADING QUANTITIES:'/)                          1990
  215 FORMAT ('+',T70,'RADIUS'/)                                           2000
  216 FORMAT ('+',T70,'GAMMA'/)                                            2010
  217 FORMAT ('+',T70,'OMEGA-S'/)                                          2020
  218 FORMAT ('+',T70,'OMEGA-THETA'/)                                      2030
  219 FORMAT ('+',T70,'DEOMEGA-S'/)                                        2040
  220 FORMAT ('+',T70,'B-STIFFNESS'/)                                      2050
  221 FORMAT ('+',T70,'D-STIFFNESS'/)                                      2060
  222 FORMAT ('+',T70,'B-PRIME'/)                                          2070
```

```
223 FORMAT ('+',T70,'D-PRIME'/)                                                 2080
224 FORMAT ('+',T70,'PRESSURE LOADING - NORMAL'/)                               2090
225 FORMAT ('+',T70,'PRESSURE LOADING - MERIDIONAL'/)                           2100
226 FORMAT ('+',T70,'PRESSURE LOADING - CIRCUMFERENTIAL'/)                      2110
227 FORMAT ('+',T70,'THERMAL LOADING'/)                                         2120
228 FORMAT ('+',T70,'THERMAL MOMENT'/)                                          2130
229 FORMAT ('+',T70,'D-THERMAL LOADING'/)                                       2140
230 FORMAT ('+',T70,'D-THERMAL MOMENT'/)                                        2150
231 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED MO          2160
    1DAL QUANTITIES:'/)                                                         2170
232 FORMAT ('0',T20,'PLOTS HAVE BEEN REQUESTED FOR THE BELOW LISTED SU          2180
    1MMED QUANTITIES:'/)                                                        2190
233 FORMAT ('+',T70,'N-S'/)                                                     2200
234 FORMAT ('+',T70,'N-THETA'/)                                                 2210
235 FORMAT ('+',T70,'N-S-THETA'/)                                               2220
236 FORMAT ('+',T70,'Q-S'/)                                                     2230
237 FORMAT ('+',T70,'M-S'/)                                                     2240
238 FORMAT ('+',T70,'M-THETA'/)                                                 2250
239 FORMAT ('+',T70,'M-S-THETA'/)                                               2260
240 FORMAT ('+',T70,'U'/)                                                       2270
242 FORMAT ('+',T70,'W'/)                                                       2280
241 FORMAT ('+',T70,'V'/)                                                       2290
243 FORMAT ('+',T70,'PHI-S'/)                                                   2300
244 FORMAT ('+',T70,'PHI-THETA'/)                                               2310
245 FORMAT ('+',T70,'PHI'/)                                                     2320
246 FORMAT (' ',4X,'NUMBER OF STATIONS----------------------',I5/5X,'N          2330
    1UMBER OF MODES--------------------------',I5/5X,'MAXIMUM NUMBER OF        2340
    2ITERATIONS------------',I5/5X,'MAXIMUM NUMBER OF TIME STEPS-------        2350
    3-----',I5/5X,'INCREMENTAL TIME-------------------------',E12.4/5X,'       2360
    4CONVERGENCE CRITERION-------------------',E12.4//)                        2370
247 FORMAT (' ',4X,'CHARACTERISTIC SHELL DIMENSION----------',E12.4/5X          2380
    1.'REFERENCE THICKNESS---------------------',E12.4/5X,'REFERENCE EL        2390
    2ASTICITY--------------------',E12.4/5X,'REFERENCE STRESS----------        2400
    3--------------',E12.4/5X,'REFERENCE TIME--------------------------        2410
    4',E12.4/5X,'POISSON''S RATIO------------------------',E12.4//)             2420
248 FORMAT (' ',T20,'CIRCUMFERENTIAL COORDINATES FOR THE PRINT RECORD,          2430
    1 IN RADIAN MEASURE, ARE:'/)                                                2440
249 FORMAT (' ',T10,6F16.10)                                                    2450
250 FORMAT (///' ',T20,'THE DATA PRINTED IS NON-DIMENSIONAL')                   2460
251 FORMAT (///' ',T20,'THE DATA PRINTED IS DIMENSIONAL')                       2470
252 FORMAT(' ',4X,'NUMBER OF STATIONS----------------------',I3,/5X,'M          2480
    1AXIMUM NUMBER OF ITERATIONS------------',I3,/5X,'NUMBER OF VIBRATI        2490
    2ON ANALYSES------------',I3,/5X,'CONVERGENCE CRITERION (EIGENVALUE         2500
    3S)-----',E12.4//)                                                          2510
     RETURN                                                                     2520
     END                                                                        2530
```

```
      SUBROUTINE STATIC                                                 2540
C*****************************************************************    2550
C     THIS SUBROUTINE IS THE MAJOR CONTROLLING ROUTINE FOR ALL STATIC  *2560
C     ANALYSIS PROBLEMS.  IT CONDUCTS INITIALIZATION, PREPARES INPUT   *2570
C     MATERIAL FOR PLOTTING IF REQ'D, DETERMINES SHELL GEOMETRY, STIFF-*2580
C     NESSES,LOADING (PHYSICAL AND/OR THERMAL), AND INITIAL CONTIDIONS.*2590
C     IT CONTROLS PROBLEM SOLUTION PROCEDURE.                          *2600
C*****************************************************************    2610
      IMPLICIT LOGICAL*1 (S)                                            2620
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS                        2630
      COMMON /IBL1/ MNMAX                                               2640
      COMMON /IBL2/ N(10),MNINIT                                        2650
C                   'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG      2660
      COMMON /IBL3/ M0,M1,M2,M3                                         2670
      COMMON /IBL4/ KMAX,KL                                             2680
      COMMON /IBL5/ IBCINL,IBCFNL                                       2690
      COMMON /IBL6/ KLL                                                 2700
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       2710
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               2720
      COMMON /IBL8/ LSTEP,ITR                                           2730
      COMMON /IBL9/ MAXM                                                2740
      COMMON /IBL10/ IFREQ,NTHMAX                                       2750
      COMMON /IBL11/ ICORFL,IPASS                                       2760
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  2770
      COMMON /IBL13/ ITRMAX,LSMAX                                       2780
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                              2790
      COMMON /BL3/  PR(10),PX(10),PT(10)                                2800
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),   2810
     1             ZF4M(4,4,10)                                         2820
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                        2830
C                   'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPOL & FNLPOL2840
      COMMON /BL6/ SOE,OSE,ALCAD                                        2850
      COMMON /BL7/  D1,S1                                               2860
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            2870
      COMMON /BL9/  EES(4,10),EL1S(4),GEES(4,10)                        2880
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                           2890
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                2900
      COMMON /BL12/ TDLI,TDEL                                           2910
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),        2920
     1              UNIT(4,4)                                           2930
      COMMON /BL14/ LAM2,LSD1C,LSD1N                                    2940
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),2950
     1              V3(10),W3(10)                                       2960
      COMMON /BL16/ EPS                                                 2970
      COMMON /BL17/ DEL                                                 2980
      COMMON /BL18/ EL1(4),ELL(4)                                       2990
      COMMON /BL19/ TH(36)                                              3000
      COMMON /BL20/ DECMX(200)                                          3010
```

```
      COMMON /BL23/ JAY(4,4),H(4,4)                                          3020
      COMMON /BL24/ DG(4,4,10),DL(4,4,10),DF(4,4,10)                         3030
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                     3040
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                         3050
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)      3060
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),        3070
     1              BXT2(10),BE2(10)                                         3080
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),     3090
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)               3100
      COMMON /BL31/ DELSQ,EXT1(10)                                           3110
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                      3120
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                                3130
      COMMON /BL100/ TEEO,$DYNMC                                             3140
      COMMON /BL101/ DELSD                                                   3150
      COMMON /BL102/ DELOAD                                                  3160
      COMMON /BL103/ MASS(200)                                               3170
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220),      3180
     1               Z3(4,220)                                               3190
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),          3200
     1               QS(10)                                                  3210
      COMMON /BL111/ ABZ,ABZC,ABZN,ABZ3,DD2                                  3220
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,      3230
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,       3240
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,          3250
     3                IPHIT,IPHI,$PLOTS,$MODAL                               3260
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),       3270
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),       3280
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),       3290
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),       3300
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),     3310
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),         3320
     6                YPHI(100),XSTATN(100)                                  3330
      COMMON /BLDATA/ TITLE,NJ,IMODE,NDIMEN,IPRINT,LCHMAX,IC                 3340
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                                    3350
C*****************************************************************         3360
      WRITE (6,8888)                                                         3370
      DELSD=DELOAD*DELCAD                                                    3380
      KL=KMAX-1                                                              3390
      KLL=KMAX-2                                                             3400
      KMAX1=KMAX+1                                                           3410
      KMAX2=KMAX+2                                                           3420
      AK=KL                                                                  3430
      SIGT(1)=SIGO*TKN                                                       3440
      SIGT(2)=SIGO/ELAST                                                     3450
      SIGC(1)=SIGO*CHAR/ELAST                                                3460
      SIGC(2)=SIGO*TKN**3/CHAR                                               3470
      IF (IBCINL.LT.0) GO TO 14                                              3480
      DO 98 I=1,4                                                            3490
```

```
      DO 98 J=1,4                                                       3500
      KKLM=J/4+1                                                        3510
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                  3520
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                  3530
   14 IF(IBCFNL.LT.0) GO TO 17                                          3540
      DO 99 I=1,4                                                       3550
      DO 99 J=1,4                                                       3560
      KKLM=J/4+1                                                        3570
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                  3580
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                  3590
   17 LAM=TKN/CHAR                                                      3600
      SOF=SIGO/ELAST                                                    3610
      CSE=.5*SCE                                                        3620
      D1=1.0-NU                                                         3630
      S1=1.0+NU                                                         3640
      LAM2=LAM**2                                                       3650
      IF(NDIMEN.LT.1) GO TO 228                                         3660
      SIGO=1.0                                                          3670
      ELAST=1.0                                                         3680
      TKN=1.0                                                           3690
      CHAR=1.0                                                          3700
  228 DO 230 M=1,MAXM                                                   3710
      N(M)=0.0                                                          3720
      PX(M)=0.0                                                         3730
      PT(M)=0.0                                                         3740
      PR(M)=0.0                                                         3750
      TT(M)=0.0                                                         3760
      MT(M)=0.0                                                         3770
      DT(M)=0.0                                                         3780
      DMT(M)=0.0                                                        3790
      MAXD(M)=0                                                         3800
      MAXS(M)=0                                                         3810
  230 MAXSY(M)=0                                                        3820
      CALL GEOM                                                         3830
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)        3840
     1       +IABS(IRADII)                                              3850
      ICHCK2=IABS(IBSTIF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)        3860
      IF (.NOT.$PLOTS) GO TO 11                                         3870
      DO 2 K=1,KMAX                                                     3880
    2 XSTATN(K)=FLOAT(K)                                                3890
      IF (ICHCK1.EQ.0) GO TO 11                                         3900
      DO 1 K=1,KMAX                                                     3910
      XRADII(K)=R(K)*CHAR                                               3920
      YGAMMA(K)=GAM(K)/CHAR                                             3930
      YOMEGS(K)=OMXI(K)/CHAR                                            3940
      YOMEGT(K)=OMT(K)/CHAR                                             3950
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)                                    3960
    1 CONTINUE                                                          3970
```

```
 11 CONTINUE                                                              3980
    DO 86 K=1,KMAX                                                        3990
 86 MASS(K)=0.                                                            4000
    WRITE(6,802)                                                          4010
    DO 978 K=1,KMAX                                                       4020
    RKK=R(K)*CHAR                                                         4030
    OMXIK=OMXI(K)/CHAR                                                    4040
    GAMK=GAM(K)/CHAR                                                      4050
    OMTK=OMT(K)/CHAR                                                      4060
    DEOMXK=DEOMX(K)/(CHAR*CHAR)                                           4070
978 WRITE(6,803) K,RKK,GAMK,OMXIK,OMTK,DEOMXK                             4080
805 MO=0                                                                  4090
    M1=0                                                                  4100
    M2=0                                                                  4110
    M3=0                                                                  4120
    ABN=CHAR/SIGO/TKN                                                     4130
    ZN=SIGO*TKN                                                           4140
    WRITE(6,112)                                                          4150
    DO 888 K=1,KMAX                                                       4160
    CALL BDB(K,B,DB,D,DD)                                                 4170
    BST=ELAST*TKN                                                         4180
    ZST=ELAST*TKN**3                                                      4190
    B=B*BST                                                               4200
    D=D*ZST                                                               4210
    DB=DB/CHAR*BST                                                        4220
    DD=DD/CHAR*ZST                                                        4230
    WRITE (6,71) K,B,D,DB,DD                                              4240
    IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                           4250
    YBSTIF(K)=B                                                           4260
    YDSTIF(K)=D                                                           4270
    YBBSTF(K)=DB                                                          4280
    YDDSTF(K)=DD                                                          4290
888 CONTINUE                                                              4300
    CALL PLOAD(1)                                                         4310
    CALL TLOAD(1)                                                         4320
    DO 889 M=1,MNMAX                                                      4330
    WRITE(6,113) N(M)                                                     4340
    WRITE(6,114)                                                          4350
    ICHCK3=IABS(IPR)+IABS(IPS)+IABS(IPT)+IABS(ITT)+IABS(IMT)              4360
   1       +IABS(IDTT)+IABS(IDMT)                                         4370
    DO 890 K=1,KMAX                                                       4380
    CALL PLOAD(K)                                                         4390
    CALL TLOAD(K)                                                         4400
    PRM=PR(M)/ABN                                                         4410
    PTM=PT(M)/ABN                                                         4420
    PXM=PX(M)/ABN                                                         4430
    TTM=TT(M)*ZN                                                          4440
    EMTM=MT(M)/CHAR*ZN*TKN*TKN                                            4450
```

```
      DTM=DT(M)/CHAR*ZN                                                 4460
      DMTM=DMT(M)*ZN*TKN*TKN/(CHAR*CHAR)                                4470
      WRITE (6,115) K,PRM,PXM,PTM,TTM,EMTM,DTM,DMTM                     4480
      IF (.NOT.$PLOTS.OR.(ICHCK3.EQ.0)) GO TO 890                       4490
      YPR(K)=PRM                                                        4500
      YPS(K)=PXM                                                        4510
      YPT(K)=PTM                                                        4520
      YTT(K)=TTM                                                        4530
      YMT(K)=EMTM                                                       4540
      YDTT(K)=DTM                                                       4550
      YDMT(K)=DMTM                                                      4560
  890 CONTINUE                                                          4570
      IF (M.EQ.1) ICHCK3=ICHCK1+ICHCK2+ICHCK3                           4580
      IF ($PLOTS.AND.(ICHCK3.GT.0)) CALL PLOT1(M)                       4590
  889 CONTINUE                                                          4600
      DELSQ=DEL**2                                                      4610
      TDLI=.5/DEL                                                       4620
      TDEL=2.0*DEL                                                      4630
      MNINIT=1                                                          4640
      MNMAXO=MNMAX                                                      4650
      DO 20 I=1,4                                                       4660
      DO 20 J=1,4                                                       4670
      UNIT(I,J)=0.0                                                     4680
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                          4690
      NMAX=MAXM*KMAX2                                                   4700
      DO 22 K=1,NMAX                                                    4710
      DO 22 I=1,4                                                       4720
      ZDOT(I,K)=0.0                                                     4730
      Z0(I,K)=0.0                                                       4740
      Z2(I,K)=0.0                                                       4750
      Z3(I,K)=0.0                                                       4760
   22 Z(I,K)=0.0                                                        4770
      ALOAD=DELOAD                                                      4780
      CALL PMATRX                                                       4790
      LSTEP=1                                                           4800
      LCHANG=0                                                          4810
      ITR=1                                                             4820
      ICOREL=0                                                          4830
      IF(MNMAX.EQ.MAXM) ICOREL=1                                        4840
      IPASS=0                                                           4850
  400 CALL XANDZ                                                        4860
      IF(ITRMAX.EQ.1) GO TO 50                                          4870
      MNMAXO=MNMAX                                                      4880
      IF(IPASS.LT.2) CALL MODES                                         4890
      IF((NCONV.EQ.1).AND.(ITR.GT.1)) GO TO 50                          4900
      IF(ITR.LT.ITRMAX) GO TO 23                                        4910
      IF(LCHANG.LT.LCHMAX) GO TO 30                                     4920
      WRITE(6,220) NO                                                   4930
```

```
      GO TO 500                                                          4940
   50 FL=LSTEP                                                           4950
      FI=IPRINT                                                          4960
      LI=LSTEP/IPRINT                                                    4970
      FLI=LI                                                             4980
      FT=FLI-FL/FI                                                       4990
      IF(FT.EQ.0.) CALL OUTPUT(IMODE)                                    5000
      IF(LSTEP.EQ.1) ITR=1                                               5010
      IF(LSTEP.EQ.1) ITRPR=1                                             5020
      IF(ITR.GT.ITRPR) ITRPR=ITR                                         5030
      IF(LSTEP.GE.LSMAX) GO TO 360                                       5040
   60 DO 61 MN=1,MNMAXO                                                  5050
      DO 61 K=1,KMAX2                                                    5060
      IK=K+(MN-1)*KMAX2                                                  5070
      DO 61 I=1,4                                                        5080
      ZN=2.0*Z(I,IK)-ZO(I,IK)                                            5090
      ZO(I,IK)=Z(I,IK)                                                   5100
   61 Z(I,IK)=ZN                                                         5110
      IF(LSTEP.GE.LSMAX) GO TO 360                                       5120
   62 ALOAD=ALOAD+DELOAD                                                 5130
      LSTEP=LSTEP+1                                                      5140
      ITR=1                                                              5150
      GO TO 400                                                          5160
  360 WRITE(6,221) NO                                                    5170
      GO TO 500                                                          5180
   23 ITR=ITR+1                                                          5190
      GO TO 400                                                          5200
   30 IF(LSTEP-1) 310,310,320                                            5210
  310 WRITE(6,223)                                                       5220
      GO TO 500                                                          5230
  320 WRITE(6,222)                                                       5240
      LCHANG=LCHANG+1                                                    5250
      LSTEP=LSTEP-1                                                      5260
      ALOAD=ALOAD-DELOAD                                                 5270
      DELOAD=DELOAD/5.0                                                  5280
      DO 32 MN=1,MNMAXO                                                  5290
      DO 32 K=1,KMAX2                                                    5300
      IK=K+(MN-1)*KMAX2                                                  5310
      DO 32 I=1,4                                                        5320
   32 Z(I,IK)=ZO(I,IK)                                                   5330
      GO TO 62                                                           5340
C*****************************************************************      5350
   71 FORMAT(20X,I3,4X,4E20.6)                                           5360
  112 FORMAT(////17X,12H  STATION    20H       B STIFFNESS    20H       D ST
     1IFFNESS    20H       B PRIME        20H       D PRIME        //)   5380
  113 FORMAT(///25X,44HPRESSURE AND TEMPERATURE COEFFICIENTS FOR N=I3,8H   5390
     1  FOLLOW//)                                                         5400
  114 FORMAT(5X,7HSTATION,3X,15H       PR        15H       PX        15H  5410
```

```
     1  PT        15H       TT        15H       MT        15H       DTT       1     5420
     25H       DMT       //)                                                        5430
  115 FORMAT(6X,I3,7X,7E15.4)                                                       5440
  220 FORMAT(1H1,80H                THE MAXIMUM NUMBER OF LOAD CHANGES HAVE BE     5450
     1EN MADE.  END PROBLEM NUMBERI4)                                               5460
  221 FORMAT(1H1,79H                THE MAXIMUM NUMBER OF LOAD STEPS HAVE BEEN     5470
     1 TAKEN.  END PROBLEM NUMBERI4)                                                5480
  222 FORMAT(1H1,119H               THE SOLUTION DID NOT CONVERGE WITHIN THE M     5490
     1AXIMUM NUMBER OF ITERATIONS.  THE LOAD FACTOR HAS BEEN DIVIDED BY             5500
     25.)                                                                           5510
  223 FORMAT(1H1,69H                THE SOLUTION DID NOT CONVERGE FOR THE FIRS     5520
     1T LOAD INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR T            5530
     2RY A SMALLER VALUE FOR DELOAD.)                                               5540
  802 FORMAT(1H1,17X,15H STATION        16H     RADIUS      16H     GAMMA          5550
     1      16H     OMEGA S     16H     OMEGA THETA16H     DEOMEGA S  //)          5560
  803 FORMAT(20X,I3, 9X,5E16.4)                                                     5570
 8888 FORMAT ('0',T20,'EXECUTING IN SUBROUTINE "STATIC"')                           5580
  500 RETURN                                                                        5590
      END                                                                           5600


      SUBROUTINE DYNAMC                                                           5610
C*****************************************************************************   5620
C     THIS SUBROUTINE IS ONE OF THE MAJOR CONTROLLING SUBROUTINES FOR  *          5630
C     ALL DYNAMIC ANALYSIS PROBLEMS.  IT OPERATES IN A FASHION SIMILAR *          5640
C     TO SUBROUTINE STATIC.                                            *          5650
C*****************************************************************************   5660
      IMPLICIT LOGICAL*1 ($)                                                      5670
      REAL*4 NU,LAM,LAM2,JAY,MT,LSD1B,LSD1N,MASS                                  5680
      COMMON /IBL1/ MNMAX                                                         5690
      COMMON /IBL2/ N(10),MNINIT                                                  5700
C                   'N' APPEARS AS 'NN' IN SUBROUTINES PLOAD & EFG                5710
      COMMON /IBL3/ M0,M1,M2,M3                                                   5720
      COMMON /IBL4/ KMAX,KL                                                       5730
      COMMON /IBL5/ IBCINL,IPCFNL                                                 5740
      COMMON /IBL6/ KLL                                                           5750
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),                 5760
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                        5770
      COMMON /IBL8/ LSTEP,ITR                                                     5780
      COMMON /IBL9/ MAXM                                                          5790
      COMMON /IBL10/ IFREC,NTHMAX                                                 5800
      COMMON /IBL11/ ICORFL,IPASS                                                 5810
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                            5820
      COMMON /IBL13/ ITRMAX,LSMAX                                                 5830
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                        5840
      COMMON /BL3/  PR(10),PX(10),PT(10)                                          5850
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),             5860
     1             ZF4M(4,4,10)                                                 5870
```

```
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                        5880
C                   'MT' APPEARS AS 'EMT' IN SUBROUTINES INLPOL & FNLPOL  5890
      COMMON /BL6/ SOE,OSE,ALOAD                                        5900
      COMMON /BL7/  D1,S1                                               5910
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            5920
      COMMON /BL9/  FFS(4,10),EL1S(4),GEES(4,10)                        5930
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                           5940
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                5950
      COMMON /BL12/ TOLI,TDEL                                           5960
      COMMON /BL13/ OMEGI(4,4),CAPLI(4,4),OMEGL(4,4),CAPLL(4,4),        5970
     1              UNIT(4,4)                                           5980
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    5990
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),6000
     1              V3(10),W3(10)                                       6010
      COMMON /BL16/ EPS                                                 6020
      COMMON /BL17/ DEL                                                 6030
      COMMON /BL18/ EL1(4),ELL(4)                                       6040
      COMMON /BL19/ TH(36)                                              6050
      COMMON /BL20/ DEOMX(200)                                          6060
      COMMON /BL23/ JAY(4,4),P(4,4)                                     6070
      COMMON /BL24/ DG(4,4,10),DL(4,4,10),DF(4,4,10)                    6080
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                6090
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    6100
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 6110
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),   6120
     1              BXT2(10),BE2(10)                                    6130
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),6140
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)          6150
      COMMON /BL31/ DELSQ,EXT1(10)                                      6160
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                 6170
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                           6180
      COMMON /BL100/ TEEC,$DYNMC                                        6190
      COMMON /BL101/ DELSD                                              6200
      COMMON /BL102/ DELOAD                                             6210
      COMMON /BL103/ MASS(200)                                          6220
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220), 6230
     1               Z3(4,220)                                          6240
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),     6250
     1               QS(10)                                             6260
      COMMON /BL111/ ABZ,ABZC,ABZN,AEZ3,CD2                             6270
      COMMON /BLPLOT/ IPADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF, 6280
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,  6290
     2                INTH,IMSTH,IOS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,     6300
     3                IPHIT,IPHI,$PLOTS,$MODAL                          6310
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),  6320
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),  6330
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),  6340
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),  6350
```

```
     4                  YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),     6360
     5                  YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),         6370
     6                  YPHI(100),XSTATN(100)                                  6380
      COMMON /BLK1/ VMASS,NN2,SVIBES                                           6390
      COMMON /BLDATA/ TITLE,NO,IMODE,NDIMEN,IPRINT,LCHMAX,IC                   6400
      DIMENSION SIGT(2),SIGC(2),TITLE(18)                                      6410
      DIMENSION VBAR(10),AVB(10)                                               6420
C*****************************************************************           6430
      WRITE (6,8888)                                                           6440
      DELSD=DELOAD*DELCAD                                                      6450
      KL=KMAX-1                                                                6460
      KLL=KMAX-2                                                               6470
      KMAX1=KMAX+1                                                             6480
      KMAX2=KMAX+2                                                             6490
      AK=KL                                                                    6500
      SIGT(1)=SIGO*TKN                                                         6510
      SIGT(2)=SIGO/ELAST                                                       6520
      SIGC(1)=SIGO*CHAR/ELAST                                                  6530
      SIGC(2)=SIGO*TKN**3/CHAR                                                 6540
      IF (IBGINL.LT.0) GO TO 14                                                6550
      DO 98 I=1,4                                                              6560
      DO 98 J=1,4                                                              6570
      KKLM=J/4+1                                                               6580
      OMEG1(I,J)=OMEG1(I,J)*SIGT(KKLM)                                         6590
   98 CAPL1(I,J)=CAPL1(I,J)*SIGC(KKLM)                                         6600
   14 IF(IBGENL.LT.0) GO TO 17                                                 6610
      DO 99 I=1,4                                                              6620
      DO 99 J=1,4                                                              6630
      KKLM=J/4+1                                                               6640
      OMEGL(I,J)=OMEGL(I,J)*SIGT(KKLM)                                         6650
   99 CAPLL(I,J)=CAPLL(I,J)*SIGC(KKLM)                                         6660
   17 LAM=TKN/CHAR                                                             6670
      SCE=SIGO/ELAST                                                           6680
      OSE=.5*SCE                                                               6690
      D1=1.0-NU                                                                6700
      S1=1.0+NU                                                                6710
      LAM2=LAM**2                                                              6720
      IF(NDIMEN.LT.1) GO TO 228                                                6730
      SIGO=1.0                                                                 6740
      ELAST=1.0                                                                6750
      TKN=1.0                                                                  6760
      CHAR=1.0                                                                 6770
  228 DO 230 M=1,MAXM                                                          6780
      N(M)=0.0                                                                 6790
      PX(M)=0.0                                                                6800
      PT(M)=0.0                                                                6810
      PR(M)=0.0                                                                6820
      TT(M)=0.0                                                                6830
```

```
      MT(M)=0.0                                                                 6840
      CT(M)=0.0                                                                 6850
      DMT(M)=0.0                                                                6860
      MAXD(M)=0                                                                 6870
      MAXS(M)=0                                                                 6880
  230 MAXSY(M)=0                                                                6890
      CALL GEOM                                                                 6900
      ICHCK1=IABS(IGAMMA)+IABS(IOMEGS)+IABS(IOMEGT)+IABS(IDEOMS)                6910
     1       +IABS(IRADII)                                                      6920
      ICHCK2=IABS(IBSTJF)+IABS(IDSTIF)+IABS(IBBSTF)+IABS(IDDSTF)                6930
      IF (.NOT.$PLOTS) GO TO 11                                                 6940
      DO 2 K=1,KMAX                                                             6950
    2 XSTATN(K)=FLOAT(K)                                                        6960
      IF (ICHCK1.EQ.0) GO TO 11                                                 6970
      DO 1 K=1,KMAX                                                             6980
      XRADII(K)=R(K)*CHAR                                                       6990
      YGAMMA(K)=GAM(K)/CHAR                                                     7000
      YOMEGS(K)=OMXI(K)/CHAR                                                    7010
      YOMEGT(K)=OMT(K)/CHAR                                                     7020
      YDEOMS(K)=DEOMX(K)/(CHAR*CHAR)                                            7030
    1 CONTINUE                                                                  7040
   11 CONTINUE                                                                  7050
  804 WRITE(6,810)                                                              7060
      TEED=TEEO                                                                 7070
      IF(NDIMEN.EQ.1) TEED=1.0                                                  7080
      DO 979 K=1,KMAX                                                           7090
      RKK=R(K)*CHAR                                                             7100
      OMXIK=OMXI(K)/CHAR                                                        7110
      GAMK=GAM(K)/CHAR                                                          7120
      OMTK=OMT(K)/CHAR                                                          7130
      DEOMXK=DEOMX(K)/(CHAR*CHAR)                                               7140
      AMSS=MASS(K)*TEED**2*ELAST*TKN/CHAR**2                                    7150
  979 WRITE(6,813) K,RKK,GAMK,OMXIK,OMTK,DEOMXK,AMSS                            7160
  805 MO=0                                                                      7170
      M1=0                                                                      7180
      M2=0                                                                      7190
      M3=0                                                                      7200
      ABN=CHAR/SIGO/TKN                                                         7210
      ZN=SIGO*TKN                                                               7220
      WRITE(6,112)                                                              7230
      DO 888 K=1,KMAX                                                           7240
      CALL BDB(K,B,DB,D,DD)                                                     7250
      BST=ELAST*TKN                                                             7260
      ZST=ELAST*TKN**3                                                          7270
      B=B*BST                                                                   7280
      D=D*ZST                                                                   7290
      DB=DB/CHAR*BST                                                            7300
      DD=DD/CHAR*ZST                                                            7310
```

```
      WRITE (6,71) K,B,D,DB,DD                                          7320
      IF (.NOT.$PLOTS.OR.(ICHCK2.EQ.0)) GO TO 888                       7330
      YBSTIF(K)=B                                                       7340
      YDSTIF(K)=D                                                       7350
      YBBSTF(K)=DB                                                      7360
      YDDSTF(K)=DD                                                      7370
  888 CONTINUE                                                          7380
      CALL PLOAD(1)                                                     7390
      CALL TLOAD(1)                                                     7400
      DELSQ=DEL**2                                                      7410
      TDLI=.5/DEL                                                       7420
      TDEL=2.0*DEL                                                      7430
      MNINIT=1                                                          7440
      MNMAXO=MNMAX                                                      7450
      DO 20 I=1,4                                                       7460
      DO 20 J=1,4                                                       7470
      UNIT(I,J)=0.0                                                     7480
   20 IF(I.EQ.J) UNIT(I,J)=1.0                                          7490
      NMAX=MAXM*KMAX2                                                   7500
      ICHCK3 = ICHCK1 + ICHCK2                                          7510
      IF($PLOTS.AND.(ICHCK3.GT.0)) CALL PLOT1(1)                        7520
      IF($VIBES) CALL VIBERS(&500)                                      7530
      DO 22 K=1,NMAX                                                    7540
      DO 22 I=1,4                                                       7550
      ZDOT(I,K)=0.0                                                     7560
      ZO(I,K)=0.0                                                       7570
      Z2(I,K)=0.0                                                       7580
      Z3(I,K)=0.0                                                       7590
   22 Z(I,K)=0.0                                                        7600
      IF(IC.EQ.0) GO TO 834                                             7610
      CALL INITL                                                        7620
      ACO=CHAR*SIGO/ELAST                                               7630
      ACM=SIGO*TKN**3/CHAR                                              7640
      DO 830 M=1,MNMAX                                                  7650
      MM=(M-1)*KMAX2                                                    7660
      WRITE(6,126) N(M)                                                 7670
      WRITE(6,127)                                                      7680
      DO 831 K=2,KMAX1                                                  7690
      MK=K+MM                                                           7700
      TU=ACO*ZO(1,MK)                                                   7710
      TV=ACO*ZO(2,MK)                                                   7720
      TW=ACO*ZO(3,MK)                                                   7730
      TM=ACM*ZO(4,MK)                                                   7740
      KK=K-1                                                            7750
  831 WRITE(6,71) KK,TU,TV,TW,TM                                        7760
      WRITE(6,129)                                                      7770
      DO 830 K=2,KMAX1                                                  7780
      ACO=CHAR*SIGO/(ELAST*TREO)                                        7790
```

```
      AMD=SIGO*TKN**3/(CHAR*TERO)                                       7800
      MK=K+MM                                                           7810
      TU=ACD*ZDOT(1,MK)                                                 7820
      TV=ACD*ZDOT(2,MK)                                                 7830
      TW=ACD*ZDOT(3,MK)                                                 7840
      TM=AMD*ZDOT(4,MK)                                                 7850
      KK=K-1                                                            7860
      WRITE(6,71) KK,TU,TV,TW,TM                                        7870
      DO 830 I=1,4                                                      7880
      Z(I,MK)=ZO(I,MK)+ZDOT(I,MK)*DELOAD                                7890
      Z2(I,MK)=ZO(I,MK)-ZDOT(I,MK)*DELOAD                               7900
  830 Z3(I,MK)=ZO(I,MK)-2.*ZDOT(I,MK)*DELOAD                            7910
  834 CONTINUE                                                          7920
      ALOAD=1.0                                                         7930
      CALL PMATRX                                                       7940
      LSTEP=1                                                           7950
      LCHANG=0                                                          7960
      ITR=1                                                             7970
      ICORFL=0                                                          7980
      IF(MNMAX.EQ.MAXM) ICORFL=1                                        7990
      IPASS=0                                                           8000
  400 CALL XANDZ                                                        8010
      IF(ITRMAX.EQ.1) GO TO 50                                          8020
      MNMAXO=MNMAX                                                      8030
      IF(IPASS.LT.2) CALL MODES                                         8040
      IF(NCONV.EQ.1) GO TO 50                                           8050
      IF(ITR.LT.ITRMAX) GO TO 23                                        8060
      GO TO 365                                                         8070
   50 FL=LSTEP                                                          8080
      FI=IPRINT                                                         8090
      LI=LSTEP/IPRINT                                                   8100
      FLI=LI                                                            8110
      FT=FLI-FL/FI                                                      8120
      IF(FT.EQ.0.) CALL OUTPUT(IMODE)                                   8130
      IF(LSTEP.EQ.1) ITR=1                                              8140
      IF(LSTEP.EQ.1) ITRPR=1                                            8150
      IF(ITR.GT.ITRPR) ITRPR=ITR                                        8160
      IF(LSTEP.GE.LSMAX) GO TO 360                                      8170
      DO 65 MN=1,MNMAXO                                                 8180
      DO 65 K=1,KMAX2                                                   8190
      IK=K+(MN-1)*KMAX2                                                 8200
      DO 65 I=1,4                                                       8210
      ZN=3.0*(Z(I,IK)-ZO(I,IK))+Z2(I,IK)                                8220
      Z3(I,IK)=Z2(I,IK)                                                 8230
      Z2(I,IK)=ZO(I,IK)                                                 8240
      ZO(I,IK)=Z(I,IK)                                                  8250
   65 Z(I,IK)=ZN                                                        8260
      ALOAD=1.0                                                         8270
```

```
      LSTEP=LSTEP+1
      ITR=1
      GO TO 400
   23 ITR=ITR+1
      GO TO 400
  360 WRITE(6,271)
      WRITE(6,188)(AVB(M),M=1,MAXM)
      WRITE(6,189) ITRPR
      GO TO 500
  365 IF(LSTEP.EQ.1) GO TO 367
      WRITE(6,266) ITRMAX,LSTEP,NO
      WRITE(6,188)(AVB(M),M=1,MAXM)
      GO TO 500
  367 WRITE(6,273)
      GO TO 500
C*****************************************************************************
   71 FORMAT(20X,I3,4X,4E20.6)
  112 FORMAT(////17X,12H  STATION    20H       B STIFFNESS    20H       D ST
     1IFFNESS    20H       D PRIME        20H       D PRIME        //)
  126 FORMAT(////5X,29HTHE INITIAL CONDITIONS FOR N=I3,8H  FOLLOW//)
  127 FORMAT(19X,7HSTATION,3X,20H       U                20H       V
     1     20H       W                20H       M S              //)
  129 FORMAT(///19X,7HSTATION,3X,20H       U DOT            20H       V DOT
     1          20H       W DOT            20H       M S DOT          //)
  188 FORMAT(////' THE MAXIMUM VBAR FOR EACH MODE IS'/10E11.4)
  189 FORMAT(/'  THE MAXIMUMUM NUMBER OF ITERATIONS TAKEN IS  ',I4)
  266 FORMAT(1H1,35H   THE SOLUTION DID NOT CONVERGE INI3,24H ITERATIONS
     1 AT TIME STEPI5,21H.  END PROBLEM NUMBERI4,1H.)
  271 FORMAT(1H1,79H              THE MAXIMUM NUMBER OF TIME STEPS HAVE BEEN
     1 TAKEN.  END PROBLEM NUMBERI4)
  273 FORMAT(1H1,69H              THE SOLUTION DID NOT CONVERGE FOR THE FIRS
     1T TIME INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPUT DATA, OR T
     2RY A SMALLER VALUE FOR DELOAD.)
  810 FORMAT(1H1, 5X,15H STATION        16H      RADIUS      16H      GAMMA
     1      16H      OMEGA S     16H      OMEGA THETA16H      DEOMEGA S  16H
     2     MASS        //)
  813 FORMAT( 8X,I3, 9X,6E16.4)
 8888 FORMAT('0',T20,'EXECUTING IN SUBROUTINE "DYNAMIC"')
  500 RETURN
      END


      SUBROUTINE VIBERS(*)
C*****************************************************************************
C     THIS IS THE MAJOR CONTROLLING SUBROUTINE FOR THE FREE VIBRATION  *
C     ANALYSIS PORTION OF THE PROGRAM. THE MODE OF OPERATION (FREQUENCY*
C     RANGE SEARCH,SUPPLIED SHIFT POINTS, MINIMUM EIGENVALUE) IS       *
C     DETERMINED AND APPROPRIATE SUBROUTINES CALLED.                   *
```

```
C*************************************************************************      8740
      IMPLICIT LOGICAL*1 ($)                                                   8750
      REAL*8 EPSVIB,EPSVEC,EPSAIT,TWOPI,WV,XV,YV                               8760
      COMMON /IBL2/ N(10),MNINIT                                               8770
      COMMON/IBL7/ MNMAXO,MAXO(10),MAXS(10),MAXSY(10),MODVIB(3,100),ITEM       8780
     1P(110)                                                                   8790
      COMMON /BL100/ TEEC,$DYNMC                                               8800
      COMMON /BL101/ DELSO                                                     8810
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)       8820
      COMMON /BLK2/ IVIBES,IVB                                                 8830
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP                                8840
      COMMON /BLK5/ $FLAG,$TAG,$DOVEC                                          8850
      COMMON /BLK8/ EPSVIB,EPSVEC,EPSAIT                                       8860
      COMMON /BLK9/ ITR1,ITR2                                                  8870
      COMMON /BLK10/ ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS               8880
C*************************************************************************      8890
      WRITE(6,9999)                                                            8900
      DELSO = 1.0                                                              8910
      IF(EPSVIB .LT. (1.0-08)) CALL VECOUT(3,&6)                               8920
    6 IF(ITR1 .LE.3) CALL VECOUT(2,&5)                                         8930
      DO 4 IVB = 1,IVIBES                                                      8940
      KEY1 = MODVIB(1,IVB)                                                     8950
      N(1) = MODVIB(2,IVB)                                                     8960
      $DOVEC = .FALSE.                                                         8970
      IF(MODVIB(3,IVB).EQ.1) $DOVEC = .TRUE.                                   8980
      IF(KEY1.EQ.4) CALL RANGER(&4)                                            8990
      IF(KEY1.GE.1 .AND. KEY1.LE.3) CALL VECOUT(6,&1)                          9000
      SHFTPT(1,IVB) = 0.0                                                      9010
      $DOVEC = .TRUE.                                                          9020
      KEY1 = 1                                                                 9030
      CALL VECOUT(5,&1)                                                        9040
    1 DO 3 J = 1,KEY1                                                          9050
      ASHIFT = -((SHFTPT(J,IVB) * SNGL(TWOPI) * TEEC)**2)                      9060
      WRITE(6,8888) ASHIFT,SHFTPT(J,IVB)                                       9070
      K = J                                                                    9080
      IF(ASHIFT .EQ. (0.0)) K = 1                                              9090
      CALL START(K)                                                            9100
      CALL PMATRX                                                              9110
      CALL ITREAL(&2)                                                          9120
      CALL ITCPLX(&3)                                                          9130
      GO TO 3                                                                  9140
    2 CALL VECOUT(9,&3)                                                        9150
    3 CONTINUE                                                                 9160
    4 CONTINUE                                                                 9170
    5 RETURN 1                                                                 9180
C*************************************************************************      9190
 8888 FORMAT('0',52X,'SHIFT POINT IS',E14.6/66X,'(',E14.6,' HZ )')             9200
 9999 FORMAT(/////51X,'EXECUTING IN SUBROUTINE VIBERS'//49X,'PERFORMING        9210
```

```
     1FREE VIBRATION ANALYSES'///)                                          9220
      END                                                                   9230

      SUBROUTINE PMATRX                                                     9240
C*****************************************************************        9250
C         THIS SUBROUTINE CALLS THE SUBROUTINES HJ(K,MN), EFG(K,MN), ABC,   9260
C         AND PANDD(K,MN) TO SET UP THE P, P-BAR AND P-HAT MATRICES GIVEN   9270
C         BY EQUATIONS (30) OF REFERENCE (2).                               *
C         INTERNALLY, MATRICES DL, DG AND DF ARE SET UP FOR THE CALCULA-*   9290
C         TION OF X(1) GIVEN BY EQUATION (31A) OF REF. (2), WHERE           *
C                                                                           *
C         X(1) = DL*SMALL-L(1) + DG*SMALL-G(1) + DF*SMALL-F(1)              *
C                                                                           *
C         THE SPECIAL P MATRIX FOR A SHELL WITH AN INITIAL POLE IS ALSO *   9340
C         COMPUTED HERE.                                                    *
C         MATRICES ZF1M,ZF2M,ZF3M,ZF4M ARE SET UP FOR THE CALCULATION OF*   9360
C         Z(K+1) GIVEN BY EQUATION (31B) OF REF. (2), WHERE                 *
C                                                                           *
C         Z(K+1)=ZF1M*SMALL-L(K) + ZF2M*X(K) + ZF3M*X(K-1) + ZF4M*SMALL-F   9390
C                                                                           *
C         IF THE SHELL HAS A FINAL POLE, THE MATRICES CL0,CL1,CL2 ARE   *   9410
C         PREPARED FOR THE CALCULATION OF Z(K)                          *   9420
C*****************************************************************        9430
      IMPLICIT LOGICAL*1 (S)                                                9440
      REAL JAY                                                              9450
      COMMON /IBL1/ MNMAX                                                   9460
      COMMON /IBL2/ N(10),MNINIT                                            9470
      COMMON /IBL3/ M0,M1,M2,M3                                             9480
      COMMON /IBL4/ KMAX,KL                                                 9490
      COMMON /IBL5/ IBCINL,IBCFNL                                           9500
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                  9510
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),       9520
     1             ZF4M(4,4,10)                                             9530
      COMMON /BL13/ OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),            9540
     1              UNIT(4,4)                                               9550
      COMMON /BL23/ JAY(4,4),G(4,4)                                         9560
      COMMON /BL24/ DG(4,4,10),DL(4,4,10),DF(4,4,10)                        9570
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                    9580
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220),     9590
     1               Z3(4,220)                                              9600
      COMMON /BLK1/ VMASS,NN2,SVIBES                                        9610
      COMMON /BLK10/ ASHIFT,ATHPUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS            9620
      DIMENSION PATA(4,4),PBTA(4,4),POTA(4,4),PJTA(4,4),DLL(4,4),PTR(4,     9630
     14),DGG(4,4),ZF1(4,4),ZF2(4,4),ZFPO(4,4),ZFP1(4,4),ZFP2(4,4),IPIVO     9640
     2T(4),INDEX(4,2),CL0(4,4),CL1(4,4),CL2(4,4),G1(4)                      9650
      EQUIVALENCE (CL0(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1)),       9660
     1(ZFPO(1),PATA(1)),(ZFP1(1),PBTA(1)),(ZFP2(1),POTA(1))                 9670
```

```
     2,(ZF1(1),DLL(1)),(ZF2(1),PTR(1))
C*****************************************************************
      ASHFUM = ASHIFT
      ATSHF = ASHFUM
      IF(IBCINL.LT.0) GO TO 10
      DO 1 MN=MNINIT,MNMAX
      CALL HJ(1,MN)
      CALL EFG(1,MN)
      CALL ABC
      CALL MATINV(C,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 3 I=1,4
      DO 3 J=1,4
      SUMA=0.
      SUMB=0.
      SUMO=0.
      SUMJ=0.
      DO 4 L=1,4
      SUMJ=SUMJ+OMEG1(I,L)*JAY(L,J)
      SUMA=SUMA+C(I,L)*A(L,J)
      SUMB=SUMB+C(I,L)*BEE(L,J)
    4 SUMO=SUMO+OMEG1(I,L)*H(L,J)
      PATA(I,J)=SUMA+UNIT(I,J)
      PBTA(I,J)=SUMB
      POTA(I,J)=SUMO
    3 PJTA(I,J)=SUMJ
      DO 5 I=1,4
      DO 5 J=1,4
      SUMOB=0.
      SUMOA=0.
      SUMOC=0.
      DO 6 L=1,4
      SUMOB=SUMOB+POTA(I,L)*PBTA(L,J)
      SUMOA=SUMOA+POTA(I,L)*PATA(L,J)
    6 SUMOC=SUMOC+POTA(I,L)*C(L,J)
      DLL(I,J)=SUMOB+PJTA(I,J)+CAPL1(I,J)
      PTR(I,J)=SUMOA
    5 DGG(I,J)=SUMOC
      CALL MATINV(DLL,4,PTR,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 1 I=1,4
      DO 1 J=1,4
      SUMD=0.
      SUME=0.
      DO 7 L=1,4
      SUMD=SUMD+DLL(I,L)*DGG(L,J)
    7 SUME=SUME-DLL(I,L)*OMEG1(L,J)
      DG(I,J,MN)=SUMD
      IF($VIBES) GO TO 2
      DF(I,J,MN)=SUME
```

```
      CL(I,J,MN)=CLL(I,J)                                              10160
    2 IJ = 1 + KMAX * (MN-1)                                           10170
    1 P(I,J,IJ)=PTR(I,J)                                               10180
      GO TO 20                                                         10190
   10 DO 11 MN=MNINIT,MNMAX                                            10200
      NN=IABS(N(MN))                                                   10210
      IJ=1+KMAX*(MN-1)                                                 10220
      DO 14 I=1,4                                                      10230
      X(I,IJ)=0.                                                       10240
      DO 14 J=1,4                                                      10250
   14 P(I,J,IJ)=0.                                                     10260
      IF(NN.GT.3) GO TO 11                                             10270
      IF(NN.GT.2) GO TO 90                                             10280
      IF(NN.GT.1) GO TO 12                                             10290
      IF(NN.GT.0) GO TO 13                                             10300
      P(3,3,IJ)=-1.                                                    10310
      P(4,4,IJ)=-1.                                                    10320
      M0=MN                                                            10330
      GO TO 11                                                         10340
   12 P(4,4,IJ)=-1.                                                    10350
      M2=MN                                                            10360
      GO TO 11                                                         10370
   90 M3=MN                                                            10380
      GO TO 11                                                         10390
   13 P(1,1,IJ)=-1.                                                    10400
      P(2,1,IJ)=1.                                                     10410
      IF(N(MN).LT.0) P(2,1,IJ)=-1.                                     10420
      M1=MN                                                            10430
   11 CONTINUE                                                         10440
   20 KLAST=KMAX                                                       10450
      IF(IBCFNL.LT.0) KLAST=KL                                         10460
      DO 23 K=2,KLAST                                                  10470
      DO 23 MN=MNINIT,MNMAX                                            10480
      CALL EFG(K,MN)                                                   10490
      CALL ABC                                                         10500
   23 CALL PANDD(K,MN)                                                 10510
      IF(IBCFNL.LT.0) GO TO 30                                         10520
      DO 40 MN=MNINIT,MNMAX                                            10530
      IKL=MN*KMAX-1                                                    10540
      JKL=KMAX*MN                                                      10550
      CALL HJ(KMAX,MN)                                                 10560
      DO 41 I=1,4                                                      10570
      DO 41 J=1,4                                                      10580
      SUMO=0.                                                          10590
      SUMP=0.                                                          10600
      SUMJ=0.                                                          10610
      DO 42 L=1,4                                                      10620
      SUMO=SUMO+OMEGL(I,L)*H(L,J)                                      10630
```

```
      SUMP=SUMP+P(I,L,IKL)*P(L,J,JKL)                                                  10640
   42 SUMJ=SUMJ+OMEGL(I,L)*JAY(L,J)                                                    10650
      PATA(I,J)=SUMO                                                                   10660
      PBTA(I,J)=UNIT(I,J)-SUMP                                                         10670
   41 PJTA(I,J)=SUMJ+CAPLL(I,J)                                                        10680
      DO 43 I=1,4                                                                      10690
      DO 43 J=1,4                                                                      10700
      SUMOP=0.                                                                         10710
      SUMJP=0.                                                                         10720
      SUMOM=0.                                                                         10730
      DO 44 L=1,4                                                                      10740
      SUMOP=SUMOP+PATA(I,L)*PBTA(L,J)                                                  10750
      SUMJP=SUMJP+PJTA(I,L)*P(L,J,JKL)                                                 10760
   44 SUMOM=SUMOM-PATA(I,L)*P(L,J,IKL)                                                 10770
      ZF1(I,J)=SUMOP-SUMJP                                                             10780
   43 ZF2(I,J)=SUMOM-PJTA(I,J)                                                         10790
      CALL MATINV(ZF1,4,ZF2,4,DETERM,IPIVOT,INDEX,4,ISCALE)                            10800
      DO 45 I=1,4                                                                      10810
      DO 45 J=1,4                                                                      10820
      SZF3=0.                                                                          10830
      SZF4=0.                                                                          10840
      DO 46 L=1,4                                                                      10850
      SZF3=SZF3+ZF1(I,L)*PATA(L,J)                                                     10860
   46 SZF4=SZF4-ZF1(I,L)*OMEGL(L,J)                                                    10870
      ZF3M(I,J,MN)=SZF3                                                                10880
      ZF4M(I,J,MN)=SZF4                                                                10890
      ZF1M(I,J,MN)=ZF1(I,J)                                                            10900
   45 ZF2M(I,J,MN)=ZF2(I,J)                                                            10910
   40 CONTINUE                                                                         10920
      RETURN                                                                           10930
   30 DO 31 MN=MNINIT,MNMAX                                                            10940
      IKL=MN*KMAX-1                                                                    10950
      NN=IABS(N(MN))                                                                   10960
      IF(NN   .GT.3) GO TO 31                                                          10970
      IF(NN.GT.2) GO TO 300                                                            10980
      IF(NN   .GT.1) GO TO 33                                                          10990
      IF(NN   .GT.0) GO TO 34                                                          11000
      MO=MN                                                                            11010
      DO 35 J=1,4                                                                      11020
      DO 35 I=1,4                                                                      11030
      CLO(I,J)=0.                                                                      11040
   35 ZFPO(I,J)=0.                                                                     11050
      ZFPO(1,1)=1.                                                                     11060
      ZFPO(2,2)=1.                                                                     11070
      ZFPO(3,1)=P(3,1,IKL)                                                             11080
      ZFPO(3,2)=P(3,2,IKL)                                                             11090
      ZFPO(3,3)=P(3,3,IKL)+1.                                                          11100
      ZFPO(3,4)=P(3,4,IKL)                                                             11110
```

```
      ZFPO(4,1)=P(4,1,IKL)                                                    11120
      ZFPO(4,2)=P(4,2,IKL)                                                    11130
      ZFPO(4,3)=P(4,3,IKL)                                                    11140
      ZFPO(4,4)=P(4,4,IKL)+1.                                                 11150
      CLO(3,3)=1.                                                             11160
      CLO(4,4)=1.                                                             11170
      CALL MATINV(ZFPO,4,CLO,4,DETERM,IPIVOT,INDEX,4,ISCALE)                  11180
      GO TO 31                                                                11190
  300 M3=MN                                                                   11200
      GO TO 31                                                                11210
   34 M1=MN                                                                   11220
      DO 60 J=1,4                                                             11230
      DO 60 I=1,4                                                             11240
      CL1(I,J)=0.                                                             11250
   60 ZFP1(I,J)=0.                                                            11260
      ZFP1(1,1)=P(1,1,IKL)+1.                                                 11270
      ZFP1(1,2)=P(1,2,IKL)                                                    11280
      ZFP1(1,3)=P(1,3,IKL)                                                    11290
      ZFP1(1,4)=P(1,4,IKL)                                                    11300
      ZFP1(2,1)=1.                                                            11310
      ZFP1(2,2)=-1.                                                           11320
      IF(N(MN).LT.0) ZFP1(2,2)=1.                                             11330
      ZFP1(3,3)=1.                                                            11340
      ZFP1(4,4)=1.                                                            11350
      CL1(1,1)=1.                                                             11360
      CALL MATINV(ZFP1,4,CL1,4,DETERM,IPIVOT,INDEX,4,ISCALE)                  11370
      GO TO 31                                                                11380
   33 M2=MN                                                                   11390
      DO 70 J=1,4                                                             11400
      DO 70 I=1,4                                                             11410
      CL2(I,J)=0.                                                             11420
   70 ZFP2(I,J)=0.                                                            11430
      ZFP2(1,1)=1.                                                            11440
      ZFP2(2,2)=1.                                                            11450
      ZFP2(3,3)=1.                                                            11460
      ZFP2(4,1)=P(4,1,IKL)                                                    11470
      ZFP2(4,2)=P(4,2,IKL)                                                    11480
      ZFP2(4,3)=P(4,3,IKL)                                                    11490
      ZFP2(4,4)=P(4,4,IKL)+1.                                                 11500
      CL2(4,4)=1.                                                             11510
      CALL MATINV(ZFP2,4,CL2,4,DETERM,IPIVOT,INDEX,4,ISCALE)                  11520
   31 CONTINUE                                                                11530
      RETURN                                                                  11540
      END                                                                     11550
```

```
      SUBROUTINE XANDZ
C*****************************************************************
C        THIS SUBROUTINE COMPUTES THE X VECTOR USING THE P, PBAR, AND  *
C        P-HAT MATRICES AND SOLVES FOR THE Z VECTOR FOR THE APPLIED AND*
C        PSUEDC LOADS.  THE SUBROUTINES PHIBET(K) AND TEAETA(K) ARE    *
C        CALLED AND THE PREVIOUS SOLUTION FOR Z, OR THE ESTIMATED VALUE*
C        OF Z IS USED TO CALCULATE THE NON-LINEAR BETA AND AETA TERMS. *
C        THE MATRICES FFS AND ELS ARE THE VALUES OF SMALL-F AT THE     *
C        INITIAL AND FINAL EDGES OF THE SHELL.  THE SUBROUTINE FORCE(K)*
C        IS CALLED TO CALCULATE THE LOAD VECTOR GEE, AND THE X VECTOR AT
C        EACH MERIDIAN STATION.  ONCE THE X VECTOR IS OBTAINED FOR ALL *
C        MERIDIAN STATIONS AND THE SOLUTION FOR Z(K+1) GIVEN BY EQUATION
C        (31B) OF REF. 2 IS OBTAINED, AND THE SOLUTION FOR Z(I) AT ALL *
C        OTHER MERIDIAN STATIONS DEFINED BY EQUATION (29B) OF REF. 2 IS*
C        OBTAINED.  THE SOLUTION Z AT THE IMAGINARY STATION OFF THE    *
C        INITIAL EDGE OF THE SHELL IS OBTAINED LAST.  THE TEST FOR CON-*
C        VERGENCE OF THE SOLUTION IS MADE AS Z IS COMPUTED.  THE SPE-  *
C        CIAL CONDITIONS FOR COMPUTING Z AT EITHER AN INITIAL POLE OR A*
C        FINAL POLE ARE ALSO  IN THIS ROUTINE.                         *
C*****************************************************************
      IMPLICIT LOGICAL=1 ($)
      REAL NU,LAM2,MT,MASS
      COMMON /IBL1/ MNMAX
      COMMON /IBL3/ M0,M1,M2,M3
      COMMON /IBL4/ KMAX,KL
      COMMON /IBL5/ IBCINL,IBCFNL
      COMMON /IBL6/ KLL
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)
      COMMON /IBL8/ LSTEP,ITR
      COMMON /IBL12/ KMAX1,KMAX2,NCONV
      COMMON /IBL13/ ITRMAX,LSMAX
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)
      COMMON /BL3/  PR(10),PX(10),PT(10)
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),
     1             ZF4M(4,4,10)
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)
      COMMON /BL6/ SOE,OSE,ALOAD
      COMMON /BL7/  D1,S1
      COMMON /BL8/  R(200),GAM(200),OMT(200)
      COMMON /BL9/  FFS(4,10),EL1S(4),GEES(4,10)
      COMMON /BL14/ LAM2,LSD1S,LSD1N
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),
     1              V3(10),W3(10)
      COMMON /BL16/ EPS
      COMMON /BL18/ EL1(4),ELL(4)
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10)
```

```
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),      12040
     1              BXT2(10),BE2(10)                                        12050
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),   12060
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)              12070
      COMMON /BL31/ DELSQ,EXT1(10)                                         12080
      COMMON /BL100/ TEEO,$DYNMC                                           12090
      COMMON /BL101/ DELSD                                                 12100
      COMMON /BL102/ DELOAD                                                12110
      COMMON /BL103/ MASS(200)                                             12120
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220),    12130
     1               Z3(4,220)                                             12140
      DIMENSION ELLS(4),FLS(4),ZT(4),IPIVOT(4),INDEX(4,2)                  12150
     1,CLO(4,4),CL1(4,4),CL2(4,4)                                          12160
     2,TZMAX(4,10),ZDD(4)                                                  12170
      EQUIVALENCE (CLO(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),ZF3M(1))       12180
C*****************************************************************       12190
      DO 201 I=1,4                                                         12200
      DO 201 M=1,MNMAX                                                     12210
      MJ=1+(M-1)*KMAX2                                                     12220
      TZMAX(I,M)=ABS(Z(I,MJ))                                              12230
      DO 201 K=2,KMAX2                                                     12240
      KM=K+(M-1)*KMAX2                                                     12250
      AZTST=ABS(Z(I,KM))                                                   12260
      IF(AZTST.GT.TZMAX(I,M))  TZMAX(I,M)=AZTST                            12270
  201 CONTINUE                                                             12280
      NCONV=1                                                              12290
      IF(ITRMAX.EQ.1) GO TO 66                                             12300
      DO 1 M=1,MNMAXO                                                      12310
      I=1+(KMAX+2)*(M-1)                                                   12320
      U1(M)=Z(1,I)                                                         12330
      V1(M)=Z(2,I)                                                         12340
      W1(M)=Z(3,I)                                                         12350
      I1=I+1                                                               12360
      U2(M)=Z(1,I1)                                                        12370
      V2(M)=Z(2,I1)                                                        12380
    1 W2(M)=Z(3,I1)                                                        12390
      IF(IBCINL.LT.0) GO TO 100                                            12400
      CALL PHIBET(1)                                                       12410
      DO 2 M=1,MNMAX                                                       12420
      BX1(M)=BX3(M)                                                        12430
      BT1(M)=BT3(M)                                                        12440
      BXT1(M)=BXT3(M)                                                      12450
    2 BE1(M)=BE3(M)                                                        12460
      CALL TEAETA(1)                                                       12470
      DO 3 M=1,MNMAX                                                       12480
      EXX1(M)=EXX3(M)                                                      12490
      ETT1(M)=ETT3(M)                                                      12500
      ETX1(M)=ETX3(M)                                                      12510
```

```
      EXT1(M)=EXT3(M)                                                   12520
      EX1(M)=EX3(M)                                                     12530
    3 ET1(M)=ET3(M)                                                     12540
  102 CALL PHIBET(2)                                                    12550
      DO 4 M=1,MNMAX                                                    12560
      BX2(M)=BX3(M)                                                     12570
      BT2(M)=BT3(M)                                                     12580
      BXT2(M)=BXT3(M)                                                   12590
    4 BE2(M)=BE3(M)                                                     12600
      CALL TEAETA(2)                                                    12610
      DO 5 M=1,MNMAX                                                    12620
      EXX2(M)=EXX3(M)                                                   12630
      ETT2(M)=ETT3(M)                                                   12640
      ETX2(M)=ETX3(M)                                                   12650
      EXT2(M)=EXT3(M)                                                   12660
      EX2(M)=EX3(M)                                                     12670
    5 ET2(M)=ET3(M)                                                     12680
      CALL PHIBET(3)                                                    12690
      CALL TEAETA(3)                                                    12700
   66 CONTINUE                                                          12710
      IF(IBCINL.LT.0) GO TO 20                                          12720
      CALL BDB(1,B1,DB,D,CD)                                            12730
      GAM1=GAM(1)                                                       12740
      CALL TLOAD(1)                                                     12750
      DO 8 M=1,MNMAX                                                    12760
      IF(ITRMAX.EQ.1) GO TO 67                                          12770
      FFS(1,M)=-TT(M)*ALOAD+OSE*(BX1(M)+BE1(M)+NU*(BT1(M)+BE1(M)))*B1   12780
      FFS(2,M)=OSF*(B1   *D1      *BXT1(M)+EX1(M)+ET1(M))               12790
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD-(EXX1(M)+ETX1(M))*SOE           12800
      GO TO 8                                                           12810
   67 FFS(1,M)=-TT(M)*ALCAD                                             12820
      FFS(2,M)=0.                                                       12830
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD                                 12840
    8 FFS(4,M)=0.                                                       12850
      DO 9 I=1,4                                                        12860
    9 EL1S(I)=ALOAD*EL1(I)                                              12870
      CALL FORCE(1)                                                     12880
   20 CALL FORCE(2)                                                     12890
      DO 10 K=3,KLL                                                     12900
      KP=K+1                                                            12910
      IF(ITRMAX.EQ.1) GO TO 10                                          12920
      CALL UPDATE                                                       12930
      CALL PHIBET(KP)                                                   12940
      CALL TEAETA(KP)                                                   12950
   10 CALL FORCE(K)                                                     12960
      IF(ITRMAX.NE.1) CALL UPDATE                                       12970
      IF(IBCFNL.LT.0) GO TO 120                                         12980
      IF(ITRMAX.EQ.1) GO TO 11                                          12990
```

```
      CALL PHIBET(KMAX)                                                 13000
      CALL TEAETA(KMAX)                                                 13010
   11 CALL FORCE(KL)                                                    13020
      CALL FORCE(KMAX)                                                  13030
      DO 12 I=1,4                                                       13040
   12 ELLS(I)=ALOAD*ELL(I)                                              13050
      CALL BDB(KMAX,BL,CB,D,DD)                                         13060
      GAML=GAM(KMAX)                                                    13070
      FLS(4)=0.                                                         13080
      CALL TLOAD(KMAX)                                                  13090
      DO 14 M=1,MNMAX                                                   13100
      IF(M.GT.1) ELLS(1)=0.0                                            13110
      IF(ITRMAX.EQ.1) GO TO 68                                          13120
      FLS(1)=-TT(M)*ALOAD+OSE*(BX3(M)+BE3(M)+NU*(BT3(M)+BE3(M)))*BL     13130
      FLS(2)   =OSE*(BL*D1*BXT3(M)+EX3(M)+ET3(M))                       13140
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD-(EXX3(M)+ETX3(M))*SOE             13150
      GO TO 69                                                          13160
   68 FLS(1)=-TT(M)*ALOAD                                               13170
      FLS(2)=0.                                                         13180
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD                                   13190
   69 CONTINUE                                                          13200
      IK=KL+KMAX*(M-1)                                                  13210
      IJ=KMAX*M                                                         13220
      L=M*KMAX2                                                         13230
      DO 14 I=1,4                                                       13240
      SUMZ=0.                                                           13250
      DO 15 J=1,4                                                       13260
C*****************************************************************     13270
C     THE FOLLOWING CARD CAUSES BOUNDARY CONS TO EXIST FOR MODE '0' ONLY  13280
C*****************************************************************     13290
      IF (M.NE.1) ELLS(J)=0                                             13300
   15 SUMZ=SUMZ+ZF1M(I,J,M)*ELLS(J)+ZF2M(I,J,M)*X(J,IJ)+ZF3M(I,J,M)*    13310
     1X(J,IK)+ZF4M(I,J,M)*FLS(J)                                        13320
   14 Z(I,L)=SUMZ                                                       13330
      LS=1                                                              13340
  150 DO 16 M=1,MNMAX                                                   13350
      DO 16 L=LS,KMAX                                                   13360
      K=KMAX2-L                                                         13370
      KPX=K-1                                                           13380
      KZ=K+1                                                            13390
      IJ=KPX+(M-1)*KMAX                                                 13400
      JK=KZ+(M-1)*KMAX2                                                 13410
      KK=JK-1                                                           13420
      DO 17 I=1,4                                                       13430
      SUMZ=0.                                                           13440
      DO 18 J=1,4                                                       13450
   18 SUMZ=SUMZ-P(I,J,IJ)*Z(J,JK)                                       13460
      SUMZ=SUMZ+X(I,IJ)                                                 13470
```

```
      ASUMZ=ABS(SUMZ)                                                    13480
      IF(ASUMZ.GT.1.E+15) ITR=ITRMAX                                     13490
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 17                     13500
      DELZ=ABS(Z(I,KK)-SUMZ)                                             13510
      ZTEST=EPS*TZMAX(I,M)                                               13520
      IF(DELZ.GT.ZTEST) NCONV=0                                          13530
   17 Z(I,KK)=SUMZ                                                       13540
   16 CONTINUE                                                           13550
      IF(IBCINL.LT.0) GO TO 30                                           13560
      DO 25 M=1,MNMAX                                                    13570
      CALL EFG(1,M)                                                      13580
      CALL ABC                                                           13590
      IJ=2+(M-1)*KMAX2                                                   13600
      IJ1=IJ+1                                                           13610
      IJ2=IJ-1                                                           13620
      DO 21 I=1,4                                                        13630
      SUMZ=0.                                                            13640
      DO 22 J=1,4                                                        13650
   22 SUMZ=SUMZ-A(I,J)*Z(J,IJ1)-BEE(I,J)*Z(J,IJ)                         13660
   21 ZT(I)=SUMZ+GEES(I,M)                                               13670
      CALL MATINV(C,4,ZT,1,DETERM,IPIVOT,INDEX,4,ISCALE)                 13680
      DO 23 I=1,4                                                        13690
   23 Z(I,IJ2)=ZT(I)                                                     13700
   25 CONTINUE                                                           13710
   30 RETURN                                                             13720
  100 CALL INLPOL                                                        13730
      DO 101 M=1,MNMAXO                                                  13740
      U1(M)=U2(M)                                                        13750
      V1(M)=V2(M)                                                        13760
      W1(M)=W2(M)                                                        13770
      IJ=3+KMAX2*(M-1)                                                   13780
      U2(M)=Z(1,IJ)                                                      13790
      V2(M)=Z(2,IJ)                                                      13800
  101 W2(M)=Z(3,IJ)                                                      13810
      GO TO 102                                                          13820
  120 IF(ITRMAX.NE.1) CALL FNLPOL                                        13830
      CALL FORCE(KL)                                                     13840
      IF(M2.EQ.0) GO TO 122                                              13850
      L=KL+(M2-1)*KMAX                                                   13860
      L1=KMAX1+(M2-1)*KMAX2                                              13870
      DO 130 I=1,4                                                       13880
      SUM=0.                                                             13890
      DO 131 J=1,4                                                       13900
  131 SUM=SUM+CL2(I,J)*X(J,L)                                            13910
      ASUMZ=ABS(SUM)                                                     13920
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TO 130                      13930
      DELZ=ABS(Z(I,L1)-SUM)                                              13940
      ZTEST=EPS*TZMAX(I,M2)                                              13950
```

```
      IF(DELZ.GT.ZTEST) NCONV=0                                          13960
  130 Z(I,L1)=SUM                                                        13970
  122 IF(M1.EQ.0) GO TO 123                                              13980
      L=KL+(M1-1)*KMAX                                                   13990
      L1=KMAX1+(M1-1)*KMAX2                                              14000
      DO 132 I=1,4                                                       14010
      SUM=0.                                                             14020
      DO 133 J=1,4                                                       14030
  133 SUM=SUM+CL1(I,J)*X(J,L)                                            14040
      ASUMZ=ABS(SUM)                                                     14050
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 132                    14060
      DELZ=ABS(Z(I,L1)-SUM)                                              14070
      ZTEST=EPS*TZMAX(I,M1)                                              14080
      IF(DELZ.GT.ZTEST) NCONV=0                                          14090
  132 Z(I,L1)=SUM                                                        14100
  123 IF(M0.EQ.0) GO TO 124                                              14110
      L=KL+(M0-1)*KMAX                                                   14120
      L1=KMAX1+(M0-1)*KMAX2                                              14130
      DO 134 I=1,4                                                       14140
      SUM=0.                                                             14150
      DO 135 J=1,4                                                       14160
  135 SUM=SUM+CL0(I,J)*X(J,L)                                            14170
      ASUMZ=ABS(SUM)                                                     14180
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-06) GO TO 134                      14190
      DELZ=ABS(Z(I,L1)-SUM)                                              14200
      ZTEST=EPS*TZMAX(I,M0)                                              14210
      IF(DELZ.GT.ZTEST) NCONV=0                                          14220
  134 Z(I,L1)=SUM                                                        14230
  124 LS=2                                                               14240
      GO TO 150                                                          14250
      END                                                                14260

      SUBROUTINE FORCE(K)                                                14270
C*************************************************************************14280
C       THIS SUBROUTINE COMPUTES THE GEE VECTOR IN EQUATION (28) OF    *  14290
C REF. 2, AND THE X VECTOR IN EQUATION (29A) OF REF. 2 FOR A GIVEN MER-*  14300
C IDIONAL STATION K.  THE VECTOR 'GEES' IS THE NONLINEAR VALUE OF 'GEE'*  14310
C AT STATION 1.                                                        *  14320
C*************************************************************************14330
      IMPLICIT LOGICAL*1 ($)                                             14340
      REAL NU,MT,LAM2,MASS,MAS                                           14350
      COMMON /IBL1/ MNMAX                                                14360
      COMMON /IBL2/ N(10),MNINIT                                         14370
      COMMON /IBL4/ KMAX,KL                                              14380
      COMMON /IBL8/ LSTEP,ITR                                            14390
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                   14400
      COMMON /IBL13/ ITRMAX,LSMAX                                        14410
```

```
      COMMON /BL3/  PR(10),PX(10),PT(10)                                14420
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),   14430
     1             ZF4M(4,4,10)                                         14440
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                        14450
      COMMON /BL6/ SOE,OSE,ALOAD                                        14460
      COMMON /BL7/  D1,S1                                               14470
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            14480
      COMMON /BL9/  FFS(4,10),EL1S(4),GEES(4,10)                        14490
      COMMON /BL11/ OMXI(200),PHEE,TO,T2                                14500
      COMMON /BL12/ TDLI,TDEL                                           14510
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    14520
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),14530
     1             V3(10),W3(10)                                        14540
      COMMON /BL17/ DEL                                                 14550
      COMMON /BL24/ DG(4,4,10),DL(4,4,10),DF(4,4,10)                    14560
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    14570
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 14580
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),   14590
     1             BXT2(10),BE2(10)                                     14600
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),14610
     1             ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)           14620
      COMMON /BL31/ DELSQ,EXT1(10)                                      14630
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                           14640
      COMMON /BL100/ TEEO,$DYNMC                                        14650
      COMMON /BL101/ DELSD                                              14660
      COMMON /BL102/ DELOAD                                             14670
      COMMON /BL103/ MASS(200)                                          14680
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220), 14690
     1             Z3(4,220)                                            14700
      DIMENSION GEE(4)                                                  14710
C*****************************************************************     14720
      FDIFF(A,B,C)=(-1.5*A+2.*B-.5*C)/DEL                               14730
      RS=R(K)                                                           14740
      RR=1./RS                                                          14750
      GA=GAM(K)                                                         14760
      CX=OMXI(K)                                                        14770
      OT=OMT(K)                                                         14780
      DL2=D1*LAM2                                                       14790
      CALL BDB(K,BS,DBS,D,DD)                                           14800
      CALL PLOAD(K)                                                     14810
      CALL TLOAD(K)                                                     14820
      MAS=MASS(K)                                                       14830
      DO 4 M=1,MNMAX                                                    14840
      IZ=K+1+(M-1)*KMAX2                                                14850
      IK=K+(M-1)*KMAX                                                   14860
      IK1=IK-1                                                          14870
      EN=N(M)                                                           14880
      ENR=EN*RR                                                         14890
```

```
      EMT=MT(M)                                                          14900
      GEE(1) =   (-PX(M)+DT(M)-DL2*GA*OX*EMT)*TDEL*ALOAD                 14910
     1   + MAS*(-5.*ZO(1,IZ )+4.*Z2(1,IZ)-Z3(1,IZ))*TDEL/DELSD           14920
      GEE(2) =   (-PT(M)-ENR*TT(M)-DL2*ENR*OT*EMT)*TDEL*ALOAD            14930
     1   + MAS*(-5.*ZO(2,IZ )+4.*Z2(2,IZ)-Z3(2,IZ))*TDEL/DELSD           14940
      GEE(3) =   (-PR(M)-(OX+OT)*TT(M)-DL2*(GA*DMT(M)-(OX*OT-ENR**2)     14950
     1*MT(M)))*TDEL*ALOAD                                                14960
     1   + MAS*(-5.*ZO(3,IZ )+4.*Z2(3,IZ)-Z3(3,IZ))*TDEL/DELSD           14970
     1 GEE(4) =  MT(M)*TDEL*ALOAD                                        14980
      IF(ITRMAX.EQ.1) GO TO 50                                           14990
      IF(K.GT.1) GO TO 6                                                 15000
      BX2T=BX1(M)                                                        15010
      BT2T=BT1(M)                                                        15020
      BXT2T=BXT1(M)                                                      15030
      BE2T=BE1(M)                                                        15040
      EXX2T=EXX1(M)                                                      15050
      ETT2T=ETT1(M)                                                      15060
      ETX2T=ETX1(M)                                                      15070
      EXT2T=EXT1(M)                                                      15080
      EX2T=EX1(M)                                                        15090
      ET2T=ET1(M)                                                        15100
      DBX= FDIFF(BX2T,BX2(M),BX3(M))                                     15110
      DBT= FDIFF(BT2T,BT2(M),BT3(M))                                     15120
      DBXT= FDIFF(BXT2T,BXT2(M),BXT3(M))                                 15130
      DBE= FDIFF(BE2T,BE2(M),BE3(M))                                     15140
      DET= FDIFF(ET2T,ET2(M),ET3(M))                                     15150
      DEX= FDIFF(EX2T,EX2(M),EX3(M))                                     15160
      DEXX= FDIFF(EXX2T,EXX2(M),EXX3(M))                                 15170
      DETX= FDIFF(ETX2T,ETX2(M),ETX3(M))                                 15180
      GO TO 7                                                            15190
    6 IF(K.LT.KMAX) GO TO 8                                              15200
      BX2T=BX3(M)                                                        15210
      BT2T=BT3(M)                                                        15220
      BXT2T=BXT3(M)                                                      15230
      BE2T=BE3(M)                                                        15240
      EXX2T=EXX3(M)                                                      15250
      ETT2T=ETT3(M)                                                      15260
      ETX2T=ETX3(M)                                                      15270
      EXT2T=EXT3(M)                                                      15280
      EX2T=EX3(M)                                                        15290
      ET2T=ET3(M)                                                        15300
      DBX=-FDIFF(BX2T,BX2(M),BX1(M))                                     15310
      DBT=-FDIFF(BT2T,BT2(M),BT1(M))                                     15320
      DBXT=-FDIFF(BXT2T,BXT2(M),BXT1(M))                                 15330
      DBE=-FDIFF(BE2T,BE2(M),BE1(M))                                     15340
      DET=-FDIFF(ET2T,ET2(M),ET1(M))                                     15350
      DEX=-FDIFF(EX2T,EX2(M),EX1(M))                                     15360
      DEXX=-FDIFF(EXX2T,EXX2(M),EXX1(M))                                 15370
```

```
      DETX=-FDIFF(ETX2T,ETX2(M),ETX1(M))                                15380
      GO TO 7                                                           15390
    8 DBX=TDLI*(BX3(M)-BX1(M))                                          15400
      DBT=TDLI*(BT3(M)-BT1(M))                                          15410
      DBE=TDLI*(BE3(M)-BE1(M))                                          15420
      DBXT=TDLI*(BXT3(M)-BXT1(M))                                       15430
      DET=TDLI*(ET3(M)-ET1(M))                                          15440
      DEX=TDLI*(EX3(M)-EX1(M))                                          15450
      DEXX=TDLI*(EXX3(M)-EXX1(M))                                       15460
      DETX=TDLI*(ETX3(M)-ETX1(M))                                       15470
      BX2T=BX2(M)                                                       15480
      BT2T=BT2(M)                                                       15490
      BXT2T=BXT2(M)                                                     15500
      BE2T=BE2(M)                                                       15510
      EXX2T=EXX2(M)                                                     15520
      ETT2T=ETT2(M)                                                     15530
      ETX2T=ETX2(M)                                                     15540
      EXT2T=EXT2(M)                                                     15550
      EX2T=EX2(M)                                                       15560
      ET2T=ET2(M)                                                       15570
    7 GEE(1)=GEE(1)-OSE*(BS*(DBX+DBE+GA*D1*(BX2T-BT2T)+NU*(DBT+DBE)     15580
     1          +ENR*D1*BXT2T)+DBS*(BX2T+BE2T+NU*(BT2T+BE2T))-2.*OX*     15590
     2            (EXX2T+ETX2T)-ENR*(EX2T+ET2T))*TDEL                    15600
      GEE(2)=GEE(2)+OSE*(BS*(ENR*(BT2T+BE2T+NU*(BX2T+BE2T))-D1*         15610
     1     (DBXT+2.*GA*BXT2T))-D1*DBS*BXT2T+2.*OT*(ETT2T+EXT2T)         15620
     2       -(DEX+DET))*TDEL                                           15630
      GEE(3)=GEE(3)+OSE*(BS*((OX+NU*OT)*(BX2T+BE2T)+(OT+NU*OX)*         15640
     1       (BT2T+BE2T))+2.*(GA*(EXX2T+ETX2T)+DEXX+DETX+ENR*           15650
     2       (EXT2T+ETT2T)))*TDEL                                       15660
   50 IF(K.GT.1) GO TO 10                                               15670
      IF(M.GT.1) EL1S(1)=0.0                                            15680
      DO 20 I=1,4                                                       15690
      GEES(I,M)=GEE(I)                                                  15700
      SUMX=0.                                                           15710
      DO 21 J=1,4                                                       15720
C*****************************************************************     15730
C     FOLLOWING CARD CAUSES A SPECIFIED BOUNDARY CONDITION VALUE TO   *  15740
C     EXISTS ONLY FOR MODE '0'                                        *  15750
C*****************************************************************     15760
      IF (M.NE.1) EL1S(J)=0                                             15770
   21 SUMX=SUMX+DL(I,J,M)*EL1S(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)*FFS(J,M)   15780
   20 X(I,IK)=SUMX                                                      15790
      GO TO 4                                                           15800
   10 DO 11 I=1,4                                                       15810
      SUMX=0.                                                           15820
      DO 12 J=1,4                                                       15830
   12 SUMX=SUMX+DEE(I,J,IK)*GEE(J)-DST(I,J,IK)*X(J,IK1)                 15840
   11 X(I,IK)=SUMX                                                      15850
```

```
    4 CONTINUE                                                          15860
      RETURN                                                            15870
      END                                                               15880

      SUBROUTINE HJ(K,MN)                                               15890
C*****************************************************************     15900
C       THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE H AND JAY        *  15910
C       MATRICES FOR BOTH BOUNDARIES OF THE SHELL                     *  15920
C*****************************************************************     15930
      REAL L2,LAM2,LSD1N,      LSD18,JAY,NU                             15940
      COMMON /IBL2/ N(10),MNINIT                                        15950
      COMMON /IBL4/ KMAX,KL                                             15960
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            15970
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                15980
      COMMON /BL14/ LAM2,LSD18,LSD1N                                    15990
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),16000
     1              V3(10),W3(10)                                       16010
      COMMON /BL17/ DEL                                                 16020
      COMMON /BL20/ DEOMX(200)                                          16030
      COMMON /BL23/ JAY(4,4),H(4,4)                                     16040
      EQUIVALENCE(L2,LAM2)                                              16050
C*****************************************************************     16060
      CALL BDB(K,B,DB,D,DD)                                             16070
      YAH=1.                                                            16080
      IF(K.EQ.1.OR.K.EQ.KMAX)YAH=2.                                     16090
      D1=(1.-NU)                                                        16100
      GA=GAM(K)                                                         16110
      OX=OMXI(K)                                                        16120
      RA=R(K)                                                           16130
      EN=N(MN)                                                          16140
      ENR=EN/RA                                                         16150
      REG=0.                                                            16160
      IF(YAH.EQ.2.) REG=1.                                              16170
      OT=OMT(K)                                                         16180
      OXT=3.*OMXI(K)-OMT(K)                                             16190
      OTX=3.*OMT(K)-OMXI(K)                                             16200
      DL=D*L2*D1*ENR                                                    16210
      H(1,1)=B                                                          16220
      H(1,2)=0.                                                         16230
      H(1,3)=0.                                                         16240
      H(1,4)=0.                                                         16250
      H(2,1)=0.                                                         16260
      H(2,2)=B*D1/2.+L2*D*D1/8.*OTX**2*REG                              16270
      H(2,3)=DL/2.*OTX*REG                                              16280
      H(2,4)=0.                                                         16290
      H(3,1)=0.                                                         16300
      H(3,2)=DL*OTX*YAH/4.                                              16310
```

```
      ENR2=ENR**2                                                       16320
      H(3,3)=L2*D*D1*(YAH*ENR2+(1.+NU)*GA**2)                           16330
      GA2=GA**2                                                         16340
      H(3,4)=L2                                                         16350
      H(4,1)=0.                                                         16360
      H(4,2)=0.                                                         16370
      H(4,3)=-1.                                                        16380
      H(4,4)=0.                                                         16390
      JAY(1,1)=NU*GA*B                                                  16400
      JAY(1,2)=NU*B*ENR                                                 16410
      JAY(1,3)=B*(OX+NU*OT)                                             16420
      JAY(1,4)=0.                                                       16430
      JAY(2,1)=-B*D1*ENR/2.-DL/8.*OXT*OTX*REG                           16440
      JAY(2,2)=-GA*H(2,2)                                               16450
      JAY(2,3)=-GA*H(2,3)                                               16460
      JAY(2,4)=0.                                                       16470
      JAY(3,1)=-L2*D*D1*((1.+NU)*GA2*OX+ENR2/4.*OXT*YAH)                16480
      JAY(3,2)=-GA*DL/2.*(    2.*OT*(1.+NU)+OTX/2.*YAH)                 16490
      JAY(3,3)=-L2*D*D1*(1.+NU+YAH)*GA*ENR2                             16500
      JAY(3,4)=L2*D1*GA                                                 16510
      JAY(4,1)=OX                                                       16520
      JAY(4,2)=0.                                                       16530
      JAY(4,3)=0.                                                       16540
      JAY(4,4)=0.                                                       16550
      DO 1 I=1,4                                                        16560
      DO 1 J=1,4                                                        16570
    1 H(I,J)=H(I,J)/2./DEL                                              16580
      RETURN                                                            16590
      END                                                               16600

      SUBROUTINE EFG(K,MN)                                              16610
C*****************************************************************     16620
C       THIS SUBROUTINE PREPARES THE ELEMENTS OF THE E, F, AND G      *  16630
C       FOR EACH MERIDIAN STATION K AND FOR EACH FOURIER MODE, MN.    *  16640
C*****************************************************************     16650
      IMPLICIT LOGICAL*1 (I)                                            16660
      REAL NU,N,     LAM2,LSD1S,LSD1N,MASS,MAS                          16670
      COMMON /IBL2/NN(10),MNINIT                                        16680
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            16690
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                16700
      COMMON /BL14/ LAM2,LSD1S,LSD1N                                    16710
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),16720
     1              V3(10),W3(10)                                       16730
      COMMON /BL20/ DECMX(200)                                          16740
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                16750
      COMMON /BL100/ TEED,SDYNMC                                        16760
      COMMON /BL101/ DELSD                                              16770
```

```
      COMMON /BL102/ DELOAD                                                    16780
      COMMON /BL103/ MASS(200)                                                 16790
      COMMON /BLK1/ VMASS,NN2,$VIBES                                           16800
C*****************************************************************************16810
      N=NN(MN)                                                                 16820
      MAS=MASS(K)                                                              16830
      VMASS = MAS                                                              16840
      IF($VIBES) MAS = 0.0                                                     16850
      CALL BDB(K,B,DB,D,DD)                                                    16860
      E(1,1)=B                                                                 16870
      E(1,2)=0.                                                                16880
      E(1,3)=0.                                                                16890
      E(1,4)=0.                                                                16900
      E(2,1)=0.                                                                16910
      D1=(1.-NU)                                                               16920
      RA=R(K)                                                                  16930
      GA=GAM(K)                                                                16940
      CX=CMXI(K)                                                               16950
      CT=CMT(K)                                                                16960
      DEX=DEOMX(K)                                                             16970
      REX=(3.*CT-CX)                                                           16980
      GA2=GA**2                                                                16990
      RXE=(3.*CX-CT)                                                           17000
      CTX=CT*CX                                                                17010
      DNLR=LAM2*D*N*D1/(2.*RA)                                                 17020
      DDNLR=DNLR*DD/D                                                          17030
      E(2,2)=B*D1/2.+LAM2*D*D1*REX**2/8.                                       17040
      E(2,3)=DNLR*REX                                                          17050
      E(2,4)=0.                                                                17060
      E(3,1)=0.                                                                17070
      E(3,2)=E(2,3)                                                            17080
      RAN=(N/RA)**2                                                            17090
      E(3,3)=LAM2*D*D1*(2.*RAN+(1.+NU)*GA2)                                    17100
      E(3,4)=LAM2                                                              17110
      E(4,1)=0.                                                                17120
      E(4,2)=0.                                                                17130
      E(4,3)=-D                                                                17140
      E(4,4)=0.                                                                17150
      F(1,1)=GA*B+DB                                                           17160
      F(1,2)=(1.+NU)*B*N/(2.*RA)+DNLR*REX*RXE/4.                               17170
      F(1,3)=B*(CX+NU*CT)+LAM2*D*D1*((1.+NU)*GA2*CX+RAN*RXE/2.)                17180
      F(1,4)=LAM2*CX                                                           17190
      F(2,1)=-F(1,2)                                                           17200
      F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*D*D1*REX/8.)*(2.*DEX-GA*(5.*CX            17210
     1-3.*CT))+LAM2*DD*D1*REX**2/8.                                            17220
      F(2,3)=DNLR*(2.*(1.+NU)*GA*CT-DEX+3.*GA*(CX-CT))+DDNLR*REX               17230
      F(2,4)=0.                                                                17240
      F(3,1)=-F(1,3)                                                           17250
```

```
      F(3,2)=DNLR*(3.*GA*CX-GA*OT*(5.+2.*NU)-DEX)+DDNLR*REX                    17260
      F(3,3)=-LAM2*D*D1*((1.+NU)*(2.*GA*CX*OT+GA**3)+2.*GA*RAN)                17270
     1+LAM2*DD*D1*((1.+NU)*GA2+2.*RAN)                                         17280
      F(3,4)=LAM2*GA*(2.-NU)                                                   17290
      F(4,1)=D*CX                                                              17300
      F(4,2)=0.                                                                17310
      F(4,3)=-D*NU*GA                                                          17320
      F(4,4)=0.                                                                17330
      G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-D1*B*RAN/2.-LAM2*D*D1*((1.+NU)*GA2*       17340
     1CX**2+RXE**2*RAN/8.)                                                     17350
     2   - 2.*MAS/DELSD                                                        17360
      G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-DNLR*2.*GA*(REX*RXE/8.          17370
     1+(1.+NU)*OTX)                                                            17380
      G(1,3)=B*(DEX+GA*(CX-OT))+DB*(CX+NU*OT)-LAM2*D*D1*GA*RAN*(RXE/2.+(       17390
     11.+NU)*CX)                                                               17400
      G(1,4)=LAM2*D1*GA*CX                                                     17410
      G(2,1)=-B*GA*N*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.*(-1.*(1.+         17420
     1NU)*GA*OTX+GA/8.*(6.*OTX-7.*CX**2-3.*OT**2)-DEX/4.*(5.*OT-3.*CX))       17430
     2-DDNLR/4.*REX*RXE                                                        17440
      G(2,2)=-GA*F(2,2)+D1/2.*B*OTX-B*RAN-LAM2*D*D1*((1.+NU)*OT**2*RAN         17450
     1-OTX/8.*REX**2)                                                          17460
     2   - 2.*MAS/DELSD                                                        17470
      G(2,3)=-B*N*(OT+NU*CX)/RA+DNLR*(GA*DEX-2.*GA2*CX-2.*(1.+NU)*OT           17480
     1*RAN+REX*(GA2+OTX))-DDNLR*REX*GA                                         17490
      G(2,4)=-NU*LAM2*OT*N/RA                                                  17500
      G(3,1)=-B*GA*(OT+NU*CX)+LAM2*D*D1*(GA*(1.+NU)*(-GA*DEX+GA2*CX            17510
     1-CX*RAN+2.*OTX*CX)+RAN/2.*(GA*CX-GA*OT-3.*DEX))                          17520
     2-LAM2*DD*D1*((1.+NU)*GA2*CX+RAN/2.*RXE)                                  17530
      G(3,2)=-B*N*(OT+NU*CX)/RA+DNLR*(2.*(1.+NU)*(OTX*OT-GA2*CX+2.*GA2         17540
     1*OT-OT*RAN)+GA*DEX+3.*GA2*(OT-CX)+OTX*REX)-DDNLR*(2.*(1.+NU)*GA          17550
     2*OT+GA*REX)                                                              17560
      G(3,3)=-B*(CX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+NU)*(OTX-RAN        17570
     1+2.*GA2)+2.*(GA2+OTX))-LAM2*DD*D1*RAN*(3.+NU)*GA                         17580
     2   - 2.*MAS/DELSD                                                        17590
      G(3,4)=-LAM2*(D1*OTX+NU*RAN)                                             17600
      G(4,1)=D*(DEX+NU*GA*CX)                                                  17610
      G(4,2)=D*NU*N*OT/RA                                                      17620
      G(4,3)=D*NU*RAN                                                          17630
      G(4,4)=-1.                                                               17640
      RETURN                                                                   17650
      END                                                                      17660

      SUBROUTINE ABC                                                           17670
C*****************************************************************************17680
C          THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE A, BEE, AND C     *  17690
C          MATRICES.                                                      *  17700
C*****************************************************************************17710
```

```
      IMPLICIT LOGICAL*1 ($)                                                17720
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                  17730
      COMMON /BL12/ TDLI,TDEL                                               17740
      COMMON /BL17/ DEL                                                     17750
      COMMON /BL25/ E(4,4),F(4,4),G(4,4)                                    17760
      COMMON /BLK1/ VMASS,NN2,$VIBES                                        17770
      COMMON /BLK10/ ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TCLDWN,TOLCLS            17780
      COMMON /BLK11/ DIAG(4)                                                17790
      DIMENSION TEMP(48)                                                    17800
      EQUIVALENCE (TEMP(1),A(1,1))                                          17810
C*****************************************************************        17820
      D2=2./DEL                                                             17830
      DO 1 I=1,4                                                            17840
      DO 1 J=1,4                                                            17850
      DEIJ=D2*E(I,J)                                                        17860
      FIJ=F(I,J)                                                            17870
      BEE(I,J)=-2.*DEIJ+TDEL*G(I,J)                                         17880
      C(I,J)=DEIJ-FIJ                                                       17890
    1 A(I,J)=DEIJ+FIJ                                                       17900
      IF(.NOT.$VIBES) GO TO 4                                               17910
      VMASS = 1.0/(TDEL * VMASS)                                            17920
      DO 2 I = 1,48                                                         17930
    2 TEMP(I) = TEMP(I) * VMASS                                             17940
      DO 3 I = 1,4                                                          17950
      BEE(I,I) = BEE(I,I) - ASHFUM * DIAG(I)                                17960
    3 CONTINUE                                                              17970
    4 RETURN                                                                17980
      END                                                                   17990


      SUBROUTINE PANDD(K,MN)                                                18000
C*****************************************************************        18010
C         THIS SUBROUTINE COMPUTES THE ELEMENTS OF THE P, P-BAR, AND    *   18020
C         P-HAT MATRICES FOR EACH MERIDIAN STATION K AND FOURIER MODE MN*   18030
C         THESE MATRICES ARE COMPUTED AND SAVED BECAUSE THEY DO NOT     *   18040
C         CHANGE DURING EITHER THE ITERATION PROCEDURE OR THE LOAD INCRE-   18050
C         MENT PROCEDURE - AS THEY ARE A FUNCTION OF THE SHELL'S INITIAL*   18060
C         GEOMETRY AND STIFFNESS.                                       *   18070
C*****************************************************************        18080
      COMMON /IBL4/ KMAX,KL                                                 18090
      COMMON /BL1/  A(4,4),BEE(4,4),C(4,4)                                  18100
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),       18110
     1             ZF4M(4,4,10)                                             18120
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                               18130
      DIMENSION TM(4,4),IPIVOT(4),INDEX(4,2),X2(4)                          18140
C*****************************************************************        18150
      IKL=K+KMAX*(MN-1)                                                     18160
      KLI=IKL-1                                                             18170
```

```
      DO 1 I=1,4                                                        18180
      DO 1 J=1,4                                                        18190
      SUM=0.                                                            18200
      DO 2 L=1,4                                                        18210
    2 SUM=SUM+C(I,L)*P(L,J,KLI)                                         18220
    1 TM(I,J)=BEE(I,J)-SUM                                              18230
      CALL MATINV(TM,4,X2,0,DETERM,IPIVOT,INDEX,4,ISCALE)               18240
      DO 5 I=1,4                                                        18250
      DO 5 J=1,4                                                        18260
      SUMA=0.                                                           18270
      SUMC=0.                                                           18280
      DO 6 L=1,4                                                        18290
      SUMA=SUMA+TM(I,L)*A(L,J)                                          18300
    6 SUMC=SUMC+TM(I,L)*C(L,J)                                          18310
      P(I,J,IKL)=SUMA                                                   18320
      DEE(I,J,IKL)=TM(I,J)                                              18330
    5 DST(I,J,IKL)=SUMC                                                 18340
      RETURN                                                            18350
      END                                                               18360
```

```
      SUBROUTINE MODES                                                  18370
C*****************************************************************     18380
C          IN THIS SUBROUTINE, ARRAYS THAT DEFINE THOSE SETS OF INDICES  *  18390
C          THAT COMBINE TO EQUAL EACH VALUE OF N IN THE PROBLEM ARE DETER*  18400
C          MINED.  MODES IS CALLED PRIOR TO THE FIRST ITERATION AND AFTER*  18410
C          EVERY ITERATION UNTIL A SPECIFIED NUMBER OF FOURIER TERMS IS  *  18420
C          REACHED.  EACH FOURIER INDEX IN THE PROBLEM IS SUBTRACTED FROM*  18430
C          ALL OTHER FOURIER INDICES AND THE RESULT IS COMPARED WITH ALL *  18440
C          FOURIER INDICES TO SEE IF THE NEW VALUE EXISTS IN THE PROGRAM.*  18450
C          IF IT DOES, THE LOCATIONS OF THE TWO INDICES THAT MADE THE COM-  18460
C          BINATION ARE STORED IN TWO SPECIAL 2-DIMENSIONAL ARRAYS, ID AND  18470
C          JD.  ONE ARGUMENT OF EACH ARRAY IS THE VALUE OF THE NEW INDEX *  18480
C          AND THE OTHER IS THE NUMBER OF COMBINATIONS OF INDICES THAT   *  18490
C          ALSO GIVE THIS VALUE OF THE INDEX.  IF THERE IS NO INDEX IN THE  18500
C          PROGRAM THAT MATCHES THE NEW ONE, THEN A NEW FOURIER TERM HAS *  18510
C          BEEN GENERATED AND WILL BE CONSIDERED IN THE NEXT ITERATION FOR  18520
C          SOLUTION.                                                     *  18530
C          THE VARIABLE MAXD STORES THE TOTAL NUMBER OF SUCH COMBINATIONS*  18540
C          FOR EACH VALUE OF THE FOURIER INDEX.  IN A SIMILAR MANNER, EACH  18550
C          INDEX IS ADDED TO EVERY OTHER INDEX AND THE SUM COMPARED WITH *  18560
C          ALL INDICES.  THIS RESULT IS STORED IN THE 2-DIMENSIONAL ARRAYS  18570
C          IS AND JS, IN THE SAME MANNER AS WAS DONE FOR THE SUBTRACTION *  18580
C          CASE.  THE VARIABLE MAXS STORES THE TOTAL NUMBER OF SUMMATION *  18590
C          COMBINATIONS FOR EACH VALUE OF THE FOURIER INDEX.             *  18600
C          A SPECIAL ROUTINE HANDLES THE CASES WHERE THE INDEX IS ADDED TO  18610
C          AND SUBTRACTED FROM ITSELF.  THE 2-DIMENSIONAL ARRAY IJS STORES  18620
C          THE LOCATION OF THE INDEX AND THE VARIABLE MAXSY STORES THE   *  18630
C          TOTAL NUMBER OF SUCH COMBINATIONS.                            *  18640
C          WITH THIS PROCEDURE, THE SERIES OF PRODUCTS THAT MAKE UP THE  *  18650
C          BETA'S AND AETA'S CONTAIN NO ZERO TERM, AND THE SUMMATION IS  *  18660
C          CARRIED OUT IN PHIBET(K) AND TEAETA(K) OVER SPECIFICALLY DE-  *  18670
C          FINED LIMITS.                                                 *  18680
C*****************************************************************     18690
      COMMON /IBL1/ MNMAX                                               18700
      COMMON /IBL2/ N(10),MNINIT                                        18710
      COMMON /IBL7/ MNMAXC,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       18720
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               18730
      COMMON /IBL9/ MAXM                                                18740
      COMMON /IBL11/ ICOREL,IPASS                                       18750
C*****************************************************************     18760
      IF(MAXM.EQ.1) RETURN                                              18770
      IF(MNINIT.GT.MAXM) RETURN                                         18780
      DO 1 MN=1,MNMAXC                                                  18790
      NMN=N(MN)                                                         18800
      NNS=MN                                                            18810
      IF(MNINIT.GT.MN) NNS=MNINIT                                       18820
```

```
      DO 1 MM=NNS,MNMAXO                                                18830
      NMM=N(MM)                                                         18840
      NTEST=IABS(NMN-NMM)                                               18850
      DO 2 MMFT=1,MNMAX                                                 18860
      IF(NTEST.EQ.N(MMFT)) GO TO 10                                     18870
    2 CONTINUE                                                          18880
      IF(ICORFL.EQ.1) GO TO 1                                           18890
      MNMAX=MNMAX+1                                                     18900
      N(MNMAX)=NTEST                                                    18910
      MMFT=MNMAX                                                        18920
      IF(MNMAX.EQ.MAXM) ICORFL=1                                        18930
   10 IF(NMN-NMM) 11,1,12                                               18940
   11 LOCD=MAXD(MMFT)+1                                                 18950
      MAXD(MMFT)=LOCD                                                   18960
      ID(LOCD,MMFT)=MM                                                  18970
      JD(LOCD,MMFT)=MN                                                  18980
      GO TO 1                                                           18990
   12 LOCD=MAXD(MMFT)+1                                                 19000
      MAXD(MMFT)=LOCD                                                   19010
      ID(LOCD,MMFT)=MN                                                  19020
      JD(LOCD,MMFT)=MM                                                  19030
    1 CONTINUE                                                          19040
      DO 301 MN=1,MNMAXO                                                19050
      NMN=N(MN)                                                         19060
      NNS=MN                                                            19070
      IF(MNINIT.GT.MN) NNS=MNINIT                                       19080
      DO 301 MM=NNS,MNMAXO                                              19090
      NMM=N(MM)                                                         19100
      NTEST=NMN+NMM                                                     19110
      DO 302 MMFT=1,MNMAX                                               19120
      IF(NTEST.EQ.N(MMFT)) GO TO 310                                    19130
  302 CONTINUE                                                          19140
      IF(ICORFL.EQ.1) GO TO 301                                         19150
      IF(MNMAX.GE.MAXM) GO TO 301                                       19160
      MNMAX=MNMAX+1                                                     19170
      N(MNMAX)=NTEST                                                    19180
      MMFT=MNMAX                                                        19190
      IF(MNMAX.GE.MAXM) ICORFL=1                                        19200
  310 IF(NMN.EQ.NMM) GO TO 360                                          19210
      LOCS=MAXS(MMFT)+1                                                 19220
      MAXS(MMFT)=LOCS                                                   19230
      IS(LOCS,MMFT)=MN                                                  19240
      JS(LOCS,MMFT)=MM                                                  19250
      GO TO 301                                                         19260
  360 IF(NMN.EQ.0) GO TO 301                                            19270
      MAXSY(MMFT)=1                                                     19280
      IJS(MMFT)=MN                                                      19290
  301 CONTINUE                                                          19300
```

```
      MNINIT=MNMAXO+1                                                    19310
      IF(ICOREL.GT.0) IPASS=IPASS+1                                      19320
      IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRX                     19330
      RETURN                                                             19340
      END                                                                19350

      SUBROUTINE POLE(K)                                                 19360
C*************************************************************         19370
C         THIS SUBROUTINE PRINTS THE SOLUTION AT AN INITIAL AND A FINAL *  19380
C         POLE.                                                         *  19390
C*************************************************************         19400
      IMPLICIT LOGICAL*1 ($)                                             19410
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                  19420
      COMMON /IBL2/ N(10),MNINIT                                         19430
      COMMON /IBL3/ M0,M1,M2,M3                                          19440
      COMMON /IBL4/ KMAX,KL                                              19450
      COMMON /IBL5/ IBCINL,IBCFNL                                        19460
      COMMON /IBL7/ MNMAXO,MAXO(10),MAXS(10),MAXSY(10),IS(10,10),        19470
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)                19480
      COMMON /IBL8/ LSTEP,ITR                                            19490
      COMMON /IBL10/ IFREQ,NTHMAX                                        19500
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                   19510
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),    19520
     1             ZF4M(4,4,10)                                          19530
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                         19540
      COMMON /BL6/ SCE,CSE,ALOAD                                         19550
      COMMON /BL7/  D1,S1                                                19560
      COMMON /BL8/  R(200),GAM(200),OMT(200)                             19570
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                            19580
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                 19590
      COMMON /BL12/ TDLI,TDEL                                            19600
      COMMON /BL14/ LAM2,LSDIE,LSDIN                                     19610
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10), 19620
     1              V3(10),W3(10)                                        19630
      COMMON /BL17/ DEL                                                  19640
      COMMON /BL19/ TH(36)                                               19650
      COMMON /BL20/ DECMX(200)                                           19660
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                     19670
      COMMON /BL31/ DELSQ,SXTI(10)                                       19680
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                  19690
      COMMON /BL100/ TEEO,$DYNMC                                         19700
      COMMON /BL101/ DELSD                                               19710
      COMMON /BL102/ DELCAD                                              19720
      COMMON /BL103/ MASS(200)                                           19730
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220),  19740
     1               Z3(4,220)                                           19750
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),      19760
```

```
     1                 QS(10)
      COMMON /BL111/ ABZ,ABZC,ABZN,ABZ3,DD2
C*****************************************************************************
      CALL BDB(K,BS,DB,DS,DD)
      IF(K.EQ.KMAX) GO TO 301
      DO 202 MN=1,MNMAXO
      U1(MN)=U2(MN)
      V1(MN)=V2(MN)
      W1(MN)=W2(MN)
      I3=3+(MN-1)*KMAX2
      I2=I3-1
      U2(MN)=Z(1,I3)
      V2(MN)=Z(2,I3)
      W2(MN)=Z(3,I3)
      PHIX(MN)=0.
      PHIT(MN)=0.0
      PHI(MN)=0.
      MX(MN)=Z(4,I2)*ABZ3
      MTH(MN)=0.
      MXT(MN)=0.
      QS(MN)=0.
      TX(MN)=0.
      TTH(MN)=0.
  202 TXT(MN)=0.
      IF(M1.EQ.0) GO TO 203
      CALL INLPOL
      PHIX(M1)=PHEE*ABZO
      PHIT(M1)=-PHEE*ABZO
      QS(M1)=0.
      CONTINUE
      IF(MO.EQ.0) GO TO 204
       TX(MO) =TO*ABZ
      TTH(MO) =TO*ABZ
      MTH(MO) =MX(MO)
  204 IF(M2.EQ.0) GO TO 205
       TX(M2) = T2*ABZ
      TTH(M2) =-T2*ABZ
      TXT(M2)=-T2*ABZ
      MTH(M2) =-MX(M2)
      MXT(M2) =MTH(M2)
      GO TO 205
  203 IF(MO.EQ.0) GO TO 206
      I3=3+(MO-1)*KMAX2
      I4=I3+1
      CALL TLOAD(1)
      TX(MO)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I3-1))*ABZ
     1        -TT(MO)*ABZ*ALOAD
      TTH(MO)=TX(MO)
```

```
      MTH(MO)=MX(MO)                                                    20250
206   IF(M2.EQ.0) GO TO 205                                             20260
      I3=3+(M2-1)*KMAX2                                                 20270
      I4=I3+1                                                           20280
      TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL                          20290
       TX(M2) = TX(M2)*ABZ                                              20300
      TTH(M2) =-TX(M2)                                                  20310
      TXT(M2) =-TX(M2)                                                  20320
      MTH(M2) =-MX(M2)                                                  20330
      MXT(M2) =-MX(M2)                                                  20340
205   RETURN                                                            20350
301   CONTINUE                                                          20360
      DO 302 MN=1,MNMAXO                                                20370
      U1(MN)=U2(MN)                                                     20380
      V1(MN)=V2(MN)                                                     20390
      W1(MN)=W2(MN)                                                     20400
      PHIX(MN)=0.                                                       20410
      PHIT(MN)=0.                                                       20420
      PHI(MN)=0.                                                        20430
      IK=KMAX1+(MN-1)*KMAX2                                             20440
      MX(MN)=Z(4,IK)*ABZ3                                               20450
      MTH(MN)=0.                                                        20460
      MXT(MN)=0.                                                        20470
      QS(MN)=0.                                                         20480
      TX(MN)=0.                                                         20490
      TTH(MN)=0.                                                        20500
302   TXT(MN)=0.                                                        20510
      IF(M1.EQ.0) GO TO 303                                             20520
      CALL FNLPOL                                                       20530
      PHIX(M1)=PHEE*ABZO                                                20540
      PHIT(M1)= PHEE*ABZO                                               20550
      QS(M1)=0.                                                         20560
      CONTINUE                                                          20570
      IF(MO.EQ.0) GO TO 304                                             20580
       TX(MO) =TO*ABZ                                                   20590
      TTH(MO) =TO*ABZ                                                   20600
      MTH(MO) =MX(MO)                                                   20610
304   IF(M2.EQ.0) GO TO 305                                             20620
       TX(M2) = T2*ABZ                                                  20630
      TTH(M2) =-T2*ABZ                                                  20640
      TXT(M2) = T2*ABZ                                                  20650
      MTH(M2) =-MX(M2)                                                  20660
      MXT(M2) =MX(M2)                                                   20670
      GO TO 305                                                         20680
303   IF(MO.EQ.0) GO TO 306                                             20690
      IKM=KMAX+(MO-1)*KMAX2                                             20700
      IM1=IKM-1                                                         20710
      CALL TLOAD(KMAX)                                                  20720
```

```
      TX(MO)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+OMXI(KMAX)*Z(3,IKM+1)    20730
     1         )*ABZ-TT(MO)*ABZ*ALOAD                                       20740
      TTH(MO) =TX(MO)                                                      20750
      MTH(MO) =MX(MO)                                                      20760
  306 IF(M2.EQ.0) GO TO 305                                                20770
      IKM=KMAX+(M2-1)*KMAX2                                                20780
      IM1=IKM-1                                                            20790
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL                          20800
      TX(M2)=TX(M2)*ABZ                                                    20810
      TTH(M2) =-TX(M2)                                                     20820
      TXT(M2) = TX(M2)                                                     20830
      MTH(M2) =-MX(M2)                                                     20840
      MXT(M2) = MX(M2)                                                     20850
  305 RETURN                                                               20860
      END                                                                  20870

      SUBROUTINE INLPOL                                                    20880
C*****************************************************************       20890
C         THIS SUBROUTINE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *  20900
C         -SUB THETA, -SUB S-THETA, AETA-SUB S-S AND -SUB THETA-S AT AN *  20910
C         INITIAL POLE.                                                 *  20920
C*****************************************************************       20930
      COMMON /IBL1/ MNMAX                                                  20940
      COMMON /IBL3/ MO,M1,M2,M3                                            20950
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                     20960
      COMMON /IBL13/ ITRMAX,LSMAX                                          20970
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)                          20980
      COMMON /BL6/ SOE,OSE,ALOAD                                           20990
      COMMON /BL7/  D1,S1                                                  21000
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                   21010
      COMMON /BL17/ DEL                                                    21020
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),      21030
     1              BXT2(10),BE2(10)                                       21040
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),   21050
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)             21060
      COMMON /BL31/ DELSQ,EXT1(10)                                         21070
      COMMON /BL104/ Z(4,220),ZOUT(4,220),X(4,200),ZO(4,220),Z2(4,220),    21080
     1               Z3(4,220)                                             21090
C*****************************************************************       21100
      DO 1 MN=1,MNMAX                                                      21110
      BX1 (MN)=0.                                                          21120
      BT1 (MN)=0.                                                          21130
      BXT1(MN)=0.                                                          21140
      BE1 (MN)=0.                                                          21150
      EX1 (MN)=0.                                                          21160
      ET1 (MN)=0.                                                          21170
      ETX1(MN)=0.                                                          21180
```

```
    1 EXX1(MN)=0.                                                          21190
      IF(M1.EQ.0) RETURN                                                   21200
      I2=2+(M1-1)*KMAX2                                                    21210
      I3=I2+1                                                              21220
      I4=I3+1                                                              21230
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z(1,I2)         21240
      BET=.5*PHEE**2                                                       21250
      IF(ITRMAX.EQ.1) BET=0.                                               21260
      T2=0.                                                                21270
      IF(M2.EQ.0) GO TO 2                                                  21280
      CALL BDB(1,B,DB,D,DD)                                                21290
      I2=2+(M2-1)*KMAX2                                                    21300
      I3=I2+1                                                              21310
      I4=I3+1                                                              21320
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*SOE*BET)        21330
      Q1=.5*PHEE*T2                                                        21340
      BX1(M2)=BET                                                          21350
      BT1(M2)=-BET                                                         21360
      BXT1(M2)=-BET                                                        21370
      ETX1(M1)=Q1                                                          21380
      IF(M3.EQ.0) GO TO 2                                                  21390
      EXX1(M3)=Q1                                                          21400
      ETX1(M3)=-Q1                                                         21410
    2 T0=0.                                                                21420
      IF(M0.EQ.0) GO TO 3                                                  21430
      BX1(M0)=BET                                                          21440
      BT1(M0)=BET                                                          21450
      CALL BDB(1,B,DB,D,DD)                                                21460
      CALL TLOAD(1)                                                        21470
      I2=2+(M0-1)*KMAX2                                                    21480
      I3=I2+1                                                              21490
      I4=I3+1                                                              21500
      T0=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,I2)    21510
     1+.5*SOE*BET)-TT(M0)=ALOAD                                            21520
    3 EXX1(M1)=PHEE*(T0+.5*T2)                                             21530
      RETURN                                                               21540
      END                                                                  21550

      SUBROUTINE FNLPOL                                                    21560
C*************************************************************************21570
C         THIS SUBROUTIAE COMPUTES THE NON-LINEAR TERMS BETA-SUB S,     *  21580
C         -SUB THETA, -SUB S-THETA, AETA-SUB S-S, AND -SUB THETA-S AT A *  21590
C         FINAL POLE.                                                   *  21600
C*************************************************************************21610
      COMMON /IBL1/ MNMAX                                                  21620
      COMMON /IBL3/ M0,M1,M2,M3                                            21630
      COMMON /IBL4/ KMAX,KL                                                21640
```

```
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  21650
      COMMON /IBL13/ ITRMAX,LSMAX                                       21660
      COMMON /BL5/  TT(10),EMT(10),DT(10),DMT(10)                       21670
      COMMON /BL6/ SOE,OSE,ALOAD                                        21680
      COMMON /BL7/  D1,S1                                               21690
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                21700
      COMMON /BL17/ DEL                                                 21710
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    21720
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 21730
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220), 21740
     1               Z3(4,220)                                          21750
C*********************************************************************** 21760
      DO 1 MN=1,MNMAX                                                   21770
      BX3 (MN)=0.                                                       21780
      BT3 (MN)=0.                                                       21790
      BXT3(MN)=0.                                                       21800
      BE3 (MN)=0.                                                       21810
      EX3 (MN)=0.                                                       21820
      ET3 (MN)=0.                                                       21830
      ETX3(MN)=0.                                                       21840
    1 EXX3(MN)=0.                                                       21850
      CALL BDB(KMAX,B,DB,D,DB)                                          21860
      IF(M1.EQ.0) RETURN                                                21870
      KM=KMAX1+(M1-1)*KMAX2                                             21880
      KM1=KM-1                                                          21890
      KM2=KM-2                                                          21900
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(KMAX)*Z(1,KM) 21910
      BET=.5*PHEE**2                                                    21920
      IF(ITRMAX.EQ.1) BET=0.                                            21930
      T2=0.                                                             21940
      IF(M2.EQ.0) GO TO 2                                               21950
      KM=KMAX1+(M2-1)*KMAX2                                             21960
      KM1=KM-1                                                          21970
      KM2=KM-2                                                          21980
      T2=8*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*SOE*BET)    21990
      Q1=.5*PHEE*T2                                                     22000
      BX3(M2)=BET                                                       22010
      BT3(M2)=-BET                                                      22020
      BXT3(M2)= BET                                                     22030
      ETX3(M1)=Q1                                                       22040
      IF(M3.EQ.0) GO TO 2                                               22050
      EXX3(M3)=Q1                                                       22060
      ETX3(M3)=-Q1                                                      22070
    2 T0=0.                                                             22080
      IF(M0.EQ.0) GO TO 3                                               22090
      CALL TLOAD(KMAX)                                                  22100
      KM=KMAX1+(M0-1)*KMAX2                                             22110
      KM1=KM-1                                                          22120
```

```
      KM2=KM-2                                                          22130
      BX3(MO)=BET                                                       22140
      BT3(MO)=BET                                                       22150
      TO=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+OMXI(KMAX)*    22160
     1Z(3,KM)+.5*SOE*BET)-TT(MO)*ALOAD                                  22170
     3 EXX3(M1)=PHEE*(TO+.5*T2)                                         22180
      RETURN                                                            22190
      END                                                               22200

      SUBROUTINE PHIBET(K)                                              22210
C*****************************************************************     22220
C         THIS SUBROUTINE CALCULATES THE PHI'S AND CARRIES OUT THE      *22230
C         MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE BETA    *22240
C         NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*22250
C         IS, JS, ID, JS,IJS, MAXS, MAXD, MAXSY ARE PREPARED IN SUB-    *22260
C         ROUTINE MODES AND USED HERE.                                  *22270
C*****************************************************************     22280
      COMMON /IBL1/ MNMAX                                               22290
      COMMON /IBL2/ N(10),MNINIT                                        22300
      COMMON /IBL4/ KMAX,KL                                             22310
      COMMON /IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       22320
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               22330
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  22340
      COMMON /IBL13/ ITRMAX,LSMAX                                       22350
      COMMON /BL6/ SOE,OSE,ALOAD                                        22360
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            22370
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                           22380
      COMMON /BL11/ OMXI(200),PHEE,TO,T2                                22390
      COMMON /BL12/ TDLI,TDEL                                           22400
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),22410
     1              V3(10),W3(10)                                       22420
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    22430
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220), 22440
     1               Z3(4,220)                                          22450
C*****************************************************************     22460
      OX=OMXI(K)                                                        22470
      CT=OMT(K)                                                         22480
      RRA=1./R(K)                                                       22490
      GA=GAM(K)                                                         22500
      KP2=K+2                                                           22510
      DO 1 M=1,MNMAXO                                                   22520
      EN=N(M)                                                           22530
      IK=KP2+(M-1)*KMAX2                                                22540
      U3(M)=Z(1,IK)                                                     22550
      V3(M)=Z(2,IK)                                                     22560
      W3(M)=Z(3,IK)                                                     22570
      PHIX(M)=-TDLI*(W3(M)-W1(M))+OX*U2(M)                              22580
```

```
      PHIT(M)=EN*W2(M)*RRA+V2(M)*OT                                      22590
    1 PHI(M)=(TDLI*(V3(M)-V1(M))+GA*V2(M)+EN*U2(M)*RRA)*.5                22600
      IF(ITRMAX.EQ.1) RETURN                                             22610
      DO 9 M=1,MNMAX                                                     22620
      SMO=0.                                                             22630
      SMT=0.                                                             22640
      SMR=0.                                                             22650
      SMF=0.                                                             22660
      IF(N(M).EQ.0) GO TO 20                                             22670
      MAXL=MAXS(M)                                                       22680
      IF(MAXL.EQ.0) GO TO 2                                              22690
      DO 3 L=1,MAXL                                                      22700
      I=IS(L,M)                                                          22710
      J=JS(L,M)                                                          22720
      SMO=SMO+PHIX(I)*PHIX(J)                                            22730
      SMT=SMT-PHIT(I)*PHIT(J)                                            22740
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                            22750
    3 SMF=SMF-PHI(I)*PHI(J)                                              22760
    2 MAXL=MAXD(M)                                                       22770
      IF(MAXL.EQ.0) GO TO 4                                              22780
      DO 5 L=1,MAXL                                                      22790
      I=ID(L,M)                                                          22800
      J=JD(L,M)                                                          22810
      SMO=SMO+PHIX(I)*PHIX(J)                                            22820
      SMT=SMT+PHIT(I)*PHIT(J)                                            22830
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)                            22840
    5 SMF=SMF+PHI(I)*PHI(J)                                              22850
    4 IF(MAXSY(M).EQ.0) GO TO 10                                         22860
      I=IJS(M)                                                           22870
      SMO=SMO+PHIX(I)**2/2.                                              22880
      SMT=SMT-PHIT(I)**2/2.                                              22890
      SMR=(SMR+PHIX(I)*PHIT(I))                                          22900
      SMF=SMF-PHI(I)**2/2.                                               22910
      GO TO 10                                                           22920
   20 DO 21 L=1,MNMAXO                                                   22930
      SMO=SMO+PHIX(L)**2                                                 22940
      SMT=SMT+PHIT(L)**2                                                 22950
   21 SMF=SMF+PHI(L)**2                                                  22960
      IF(M.GT.MNMAXO) GO TO 11                                           22970
      SMO=SMO+PHIX(M)**2                                                 22980
   11 BX3(M)=SMO*.5                                                      22990
      BT3(M)=SMT*.5                                                      23000
      BE3(M)=SMF*.5                                                      23010
      BXT3(M)=0.                                                         23020
      GO TO 9                                                            23030
   10 BX3(M)=SMO                                                         23040
      BT3(M)=SMT                                                         23050
      BXT3(M)=SMR*.5                                                     23060
```

```
      BE3(M)=SMF                                                        23070
    9 CONTINUE                                                          23080
      RETURN                                                            23090
      END                                                               23100

      SUBROUTINE TEAETA(K)                                              23110
C*****************************************************************    23120
C       THIS SUBROUTINE CALCULATES THE INPLANE FORCES AND CARRIES OUT *  23130
C       THE MULTIPLYING AND SUMMATION PROCEDURE FOR COMPUTING THE AETA*  23140
C       NON-LINEAR TERMS FOR A GIVEN MERIDIONAL STATION K.  THE ARRAYS*  23150
C       IS, JS, ID, JS, IJS, MAXS, MAXD, MAXSY PREPARED IN SUBROUTINE *  23160
C       MODES ARE USED HERE                                           *  23170
C*****************************************************************    23180
      REAL NU,MT                                                        23190
      COMMON /IBL1/ MNMAX                                               23200
      COMMON /IBL2/ N(10),MNINIT                                        23210
      COMMON /IBL7/ MNMAXC,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       23220
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               23230
      COMMON /IBL8/ LSTEP,ITS                                           23240
      COMMON /IBL13/ ITRMAX,LSMAX                                       23250
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                        23260
      COMMON /BL6/ SOE,OSE,ALOAD                                        23270
      COMMON /BL7/  D1,S1                                               23280
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            23290
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                           23300
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                23310
      COMMON /BL12/ TDLI,TOEL                                           23320
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),23330
     1              V3(10),W3(10)                                       23340
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    23350
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 23360
      DIMENSION TX(10),TTH(10),TXT(10)                                  23370
C*****************************************************************    23380
      BRA=1./R(K)                                                       23390
      GA=GAM(K)                                                         23400
      OX=OMXI(K)                                                        23410
      OT=OMT(K)                                                         23420
      CALL BCB(K,BS,DB,DS,DD)                                           23430
      DO 1 M=1,MNMAXD                                                   23440
      EN=N(M)                                                           23450
      CALL TLOAD(K)                                                     23460
      TTS=TT(M)*ALOAD                                                   23470
      EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+OSE*(BX3(M)+BE3(M))                23480
      ET= EN *V2(M)*BRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M))           23490
      EXT=.5*(TDLI*(V3(M)-V1(M))- EN *U2(M)*BRA-GA*V2(M))+OSE*BXT3(M)   23500
      TX(M)=BS*(EX+NU*ET)-TTS                                           23510
      TTH(M)=BS*(ET+NU*EX)-TT                                           23520
```

```
 1  TXT(M)=BS*D1*EXT                                                    23530
    DO 9 M=1,MNMAX                                                      23540
    SMF=0.                                                              23550
    SMS=0.                                                              23560
    SMV=0.                                                              23570
    SME=0.                                                              23580
    SMN=0.                                                              23590
    SMT=0.                                                              23600
    IF(N(M).EQ.0) GO TO 20                                              23610
    MAXL=MAXS(M)                                                        23620
    IF(MAXL.EQ.0) GO TO 2                                               23630
    DO 3 L=1,MAXL                                                       23640
    I=IS(L,M)                                                           23650
    J=JS(L,M)                                                           23660
    SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                 23670
    SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                               23680
    SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)                               23690
    SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                               23700
    SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)                                   23710
 3  SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)                                 23720
 2  MAXL=MAXD(M)                                                        23730
    IF(MAXL.EQ.0) GO TO 4                                               23740
    DO 5 L=1,MAXL                                                       23750
    I=ID(L,M)                                                           23760
    J=JD(L,M)                                                           23770
    SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)                                 23780
    SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)                               23790
    SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)                               23800
    SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)                               23810
    SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)                                   23820
 5  SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)                                 23830
 4  IF(MAXSY(M).EQ.0) GO TO 10                                          23840
    I=IJS(M)                                                            23850
    SMF=SMF+TX(I)*PHIX(I)                                               23860
    SMS=SMS+TTH(I)*PHIT(I)                                              23870
    SMV=SMV-PHIT(I)*TXT(I)                                              23880
    SME=SME+PHIX(I)*TXT(I)                                              23890
    SMN=SMN+TX(I)*PHI(I)                                                23900
    SMT=SMT+TTH(I)*PHI(I)                                               23910
    GO TO 10                                                            23920
20  DO 21 L=1,MNMAXO                                                    23930
    SMF=SMF+TX(L)*PHIX(L)                                               23940
21  SMV=SMV+PHIT(L)*TXT(L)                                              23950
    IF(M.GT.MNMAXO) GO TO 10                                            23960
    SMF=SMF+TX(M)*PHIX(M)                                               23970
10  EXX3(M)=SMF*.5                                                      23980
    ETT3(M)=SMS*.5                                                      23990
    ETX3(M)=SMV*.5                                                      24000
```

```
      EXT3(M)=SME*.5                                                    24010
      EX3(M)=SMN*.5                                                     24020
      ET3(M)=SMT*.5                                                     24030
    9 CONTINUE                                                          24040
      DO 30 M=1,MMMAXO                                                  24050
      U1(M)=U2(M)                                                       24060
      V1(M)=V2(M)                                                       24070
      W1(M)=W2(M)                                                       24080
      U2(M)=U3(M)                                                       24090
      V2(M)=V3(M)                                                       24100
   30 W2(M)=W3(M)                                                       24110
      RETURN                                                            24120
      END                                                               24130

      SUBROUTINE UPDATE                                                 24140
C*************************************************************************24150
C        THIS SUBROUTINE UPDATES THE STORAGE LOCATIONS OF THE BETA'S AND 24160
C        ETA'S.  IT IS CALLED IN SUBROUTINE XANDZ AFTER A MERIDIAN      *24170
C        STATION CHANGE.                                                *24180
C*************************************************************************24190
      COMMON /IBL1/ MMMAX                                               24200
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                    24210
      COMMON /BL28/ EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(10) 24220
      COMMON /BL29/ BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10),   24230
     1              BXT2(10),BE2(10))                                   24240
      COMMON /BL30/ EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(10),24250
     1              ETT2(10),ETX2(10),EXT2(10),EX2(10),ET2(10)          24260
      COMMON /BL31/ DELSQ,EXT1(10)                                      24270
C*************************************************************************24280
      DO 1 M=1,MMMAX                                                    24290
      BX1(M)=BX2(M)                                                     24300
      BT1(M)=BT2(M)                                                     24310
      BXT1(M)=BXT2(M)                                                   24320
      BE1(M)=BE2(M)                                                     24330
      BX2(M)=BX3(M)                                                     24340
      BT2(M)=BT3(M)                                                     24350
      BXT2(M)=BXT3(M)                                                   24360
      BE2(M)=BE3(M)                                                     24370
      EXX1(M)=EXX2(M)                                                   24380
      ETT1(M)=ETT2(M)                                                   24390
      ETX1(M)=ETX2(M)                                                   24400
      EXT1(M)=EXT2(M)                                                   24410
      EX1(M)=EX2(M)                                                     24420
      ET1(M)=ET2(M)                                                     24430
      EXX2(M)=EXX3(M)                                                   24440
      ETT2(M)=ETT3(M)                                                   24450
      ETX2(M)=ETX3(M)                                                   24460
```

```
      EXT2(M)=EXT3(M)                                                   24470
      EX2(M)=EX3(M)                                                     24480
     1 ET2(M)=ET3(M)                                                    24490
      RETURN                                                            24500
      END                                                               24510

      SUBROUTINE OUTPUT(IMODE)                                          24520
C*****************************************************************    24530
C       THIS SUBROUTINE PREPARES THE PRINTOUT MATERIAL.  EVERY IPRINT *  24540
C       CONVERGED SOLUTION IS PRINTED.  THE FOURIER COEFFICIENTS OF THE 24550
C       INPLANE FORCES, MERIDIONAL TRANSVERSE FORCE, CIRCUMFERENTIAL  * 24560
C       BENDING MOMENT, TWISTING MOMENT AND ROTATIONS CAN BE COMPUTED * 24570
C       AND PRINTED WITH THE SOLUTION Z FOR THE FOURIER COEFFICIENTS  * 24580
C       OF THE THREE DISPLACEMENTS AND MERIDIONAL BENDING MOMENT.  THIS 24590
C       OUTPUT MATERIAL IS CONVERTED FROM DIMENSIONLESS FORM TO DIMEN-* 24600
C       SIONAL FORM HERE.  PROVISION IS MADE TO PRINT ONLY AT STATIONS* 24610
C       1, IFREQ+1, 2IFREQ+1, ETC.  THIS SUBROUTINE ALSO PERFORMS THE * 24620
C       SUMMATION PROCESS FOR COMPUTING THE TOTAL VALUES OF THE FORCES, 24630
C       MOMENTS, DISPLACEMENTS AND ROTATIONS AT THE NTHMAX POSITIONS  * 24640
C       AROUND THE CIRCUMFERENCE PRESCRIBED IN THE INPUT DATA.        * 24650
C     DATA TO BE PLOTTED IS PREPARED HERE, IF REQ'D                   * 24660
C*****************************************************************    24670
      IMPLICIT LOGICAL*1 ($)                                            24680
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS                 24690
      COMMON /IBL2/ N(10),MNINIT                                        24700
      COMMON /IBL3/ M0,M1,M2,M3                                         24710
      COMMON /IBL4/ KMAX,KL                                             24720
      COMMON /IBL5/ IBCINL,IBCFNL                                       24730
      COMMON /IBL7/ MNMAXC,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),       24740
     1              JS(10,10),ID(10,10),JD(10,10),IJS(10)               24750
      COMMON /IBL8/ LSTEP,ITR                                           24760
      COMMON /IBL10/ IFREQ,NTHMAX                                       24770
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  24780
      COMMON /IBL13/ ITRMAX,LSMAX                                       24790
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),   24800
     1             ZF4M(4,4,10)                                         24810
      COMMON /BL5/  TT(10),MT(10),DT(10),DMT(10)                        24820
      COMMON /BL6/ SOE,OSE,ALOAD                                        24830
      COMMON /BL7/  D1,S1                                               24840
      COMMON /BL8/  R(200),GAM(200),OMT(200)                            24850
      COMMON /BL10/ PHIX(10),PHIT(10),PHI(10)                           24860
      COMMON /BL11/ OMXI(200),PHEE,T0,T2                                24870
      COMMON /BL12/ TOLI,TOEL                                           24880
      COMMON /BL14/ LAM2,LSD1S,LSD1N                                    24890
      COMMON /BL15/ NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),24900
     1              V3(10),W3(10)                                       24910
      COMMON /BL17/ DEL                                                 24920
```

```
      COMMON /BL19/ TH(36)                                                      24930
      COMMON /BL20/ DECMX(200)                                                  24940
      COMMON /BL27/ BX3(10),BT3(10),BXT3(10),BE3(10)                            24950
      COMMON /BL31/ DELSQ,EXT1(10)                                              24960
      COMMON /BL32/ TKN,ELAST,CHAR,SIGO                                         24970
      COMMON /BL100/ TEEO,$DYNMC                                                24980
      COMMON /BL101/ DELSD                                                      24990
      COMMON /BL102/ DELCAD                                                     25000
      COMMON /BL103/ MASS(200)                                                  25010
      COMMON /BL104/ Z(4,220),ZDOT(4,220),X(4,200),ZO(4,220),Z2(4,220),         25020
     1               Z3(4,220)                                                  25030
      COMMON /BL110/ TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),             25040
     1               QS(10)                                                     25050
      COMMON /BL111/ ABZ,ABZO,ABZN,ABZ3,DD2                                     25060
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,         25070
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,          25080
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,             25090
     3                IPHIT,IPHI,$PLOTS,$MODAL                                  25100
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),          25110
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),          25120
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),          25130
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),          25140
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),        25150
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),            25160
     6                YPHI(100),XSTATN(100)                                     25170
      COMMON /BLK1/ VMASS,NN2,$VIBES                                            25180
      DIMENSION PTF(200),PF(200)                                                25190
C*****************************************************************************25200
      ABZO=SIGO/ELAST                                                           25210
      IF($VIBES) GO TO 182                                                      25220
      IF ($DYNMC) GO TO 181                                                     25230
      WRITE(6,101) LSTEP,ALOAD,ITR                                              25240
      GO TO 182                                                                 25250
  181 TI=LSTEP*DELOAD                                                           25260
      DTI=TI*TEEO                                                               25270
      WRITE(6,151) LSTEP,TI,DTI,ITR                                             25280
  182 LAM=TKN/CHAR                                                              25290
      ENL=1.                                                                    25300
      ABZ=SIGO*TKN                                                              25310
      ABZ3=ABZ*TKN*TKN/CHAR                                                     25320
      ABZN=CHAR*SIGO/ELAST                                                      25330
      IF(ITRMAX.EQ.1) ENL=0.                                                    25340
      DD2=1.-NU**2                                                              25350
      D2I=1./DD2                                                                25360
      DPI=1./S1                                                                 25370
      DNI=1./D1                                                                 25380
      TDLSQI=.5/DELSQ                                                           25390
      ICHCK1=IABS(INS)+IABS(INTH)+IABS(INSTH)+IABS(IQS)+IABS(IMS)               25400
```

```
     1       +IABS(IMTH)+IABS(IMSTH)                                                  25410
      ICHCK2=IABS(IU)+IABS(IV)+IABS(IW)+IABS(IPHIS)+IABS(IPHIT)                       25420
     1       +IABS(IPHI)                                                              25430
      IF(NTHMAX.EQ.0) GO TO 991                                                       25440
      IF($VIBES) GO TO 991                                                            25450
      DO 21 NTH=1,NTHMAX                                                              25460
      DO 1 MN=1,MNMAXO                                                                25470
      I1=1+(MN-1)*KMAX2                                                               25480
      I2=I1+1                                                                         25490
      U1(MN)=Z(1,I1)                                                                  25500
      U2(MN)=Z(1,I2)                                                                  25510
      V1(MN)=Z(2,I1)                                                                  25520
      V2(MN)=Z(2,I2)                                                                  25530
      W1(MN)=Z(3,I1)                                                                  25540
    1 W2(MN)=Z(3,I2)                                                                  25550
      THET=TH(NTH)                                                                    25560
      WRITE(6,116) THET                                                               25570
      DO 121 K=1,KMAX                                                                 25580
      K1=K+1                                                                          25590
      CALL BDB(K,BS,DB,DS,DD)                                                         25600
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                                         25610
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 999                                            25620
      IF(K.EQ.KMAX.AND.IBCENL.LT.0) CALL POLE(K)                                      25630
      IF(K.EQ.KMAX.AND.IBCENL.LT.0) GO TO 999                                         25640
      CALL PHIBET(K)                                                                  25650
      DEX=DEOMX(K)                                                                    25660
      RRA=1./R(K)                                                                     25670
      OX=OMXI(K)                                                                      25680
      OT=OMT(K)                                                                       25690
      GA=GAM(K)                                                                       25700
      DOXT=OX-OT                                                                      25710
      GDO=GA*DOXT                                                                     25720
      DD2D=DD2*DS                                                                     25730
      DO 3 MN=1,MNMAXO                                                                25740
      EN=N(MN)                                                                        25750
      ENR=EN*RRA                                                                      25760
      CALL TLOAD(K)                                                                   25770
      TTS=TT(MN)*ALOAD                                                                25780
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))                     25790
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))                     25800
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SOE*BXT3(MN)              25810
     1)                                                                               25820
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                                     25830
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)               25840
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DCXT*PHI(MN))                         25850
      TX(MN)=BS*(EX+NU*ET)-TTS                                                        25860
      TTH(MN)=BS*(ET+NU*EX)-TTS                                                       25870
      TXT(MN)=BS*D1*EXT                                                               25880
```

```
      MK1=K1+(MN-1)*KMAX2                                                    25890
      MX(MN)=Z(4,MK1)                                                        25900
      MTH(MN)=NU*MX(MN)+DD2D*KT-D1*MT(MN)*ALOAD                              25910
      MXT(MN)=DS*D1*KXT                                                      25920
      MK11=MK1+1                                                             25930
      MKK1=MK1-1                                                             25940
      QS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MK11)-Z(4,MKK1))*TDLI             25950
     1      +ENR*MXT(MN)-GA*MTH(MN))                                         25960
      MX(MN)=MX(MN)*ABZ3                                                     25970
      MTH(MN)=MTH(MN)*ABZ3                                                   25980
      MXT(MN)=MXT(MN)*ABZ3                                                   25990
      TX(MN)=TX(MN)*ABZ                                                      26000
      TTH(MN)=TTH(MN)*ABZ                                                    26010
      TXT(MN)=TXT(MN)*ABZ                                                    26020
      PHIX(MN)=PHIX(MN)*ABZO                                                 26030
      PHIT(MN)=PHIT(MN)*ABZO                                                 26040
      PHI(MN)=PHI(MN)*ABZO                                                   26050
      U1(MN)=U2(MN)                                                          26060
      U2(MN)=U3(MN)                                                          26070
      V1(MN)=V2(MN)                                                          26080
      V2(MN)=V3(MN)                                                          26090
      W1(MN)=W2(MN)                                                          26100
    3 W2(MN)=W3(MN)                                                          26110
      FK=K-1                                                                 26120
      FIFREQ=IFREQ                                                           26130
      KTST=(K-1)/IFREQ                                                       26140
      FKTST=KTST                                                             26150
      FKTEST=FK/FIFREQ-FKTST                                                 26160
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999                                      26170
      IF(FKTEST.NE.0.) GO TO 2                                               26180
  999 X(1,K)=0.                                                              26190
      X(2,K)=0.                                                              26200
      X(3,K)=0.                                                              26210
      X(4,K)=0.                                                              26220
      PTF(K)=0.                                                              26230
      PF(K)=0.                                                               26240
      AMX=0.                                                                 26250
      AMTH=0.                                                                26260
      AMXTH=0.                                                               26270
      ANX=0.                                                                 26280
      ANTH=0.                                                                26290
      ANXTH=0.                                                               26300
      AQS=0.                                                                 26310
      DO 72 MN=1,MNMAXO                                                      26320
      EN=N(MN)                                                               26330
      EC=EN*THET                                                             26340
      SN=SIN(EC)                                                             26350
      CS=COS(EC)                                                             26360
```

```
      X(1,K)=X(1,K)+U1(MN)*CS*ABZN                                          26370
      X(2,K)=X(2,K)+V1(MN)*SN*ABZN                                          26380
      X(3,K)=X(3,K)+W1(MN)*CS*ABZN                                          26390
      X(4,K)=X(4,K)+PHIX(MN)*CS                                             26400
      PTF(K)=PTF(K)+PHIT(MN)*SN                                             26410
      AMX=AMX+MX(MN)*CS                                                     26420
      AMTH=AMTH+MTH(MN)*CS                                                  26430
      AMXTH=AMXTH+MXT(MN)*SN                                                26440
      ANX=ANX+TX(MN)*CS                                                     26450
      ANTH=ANTH+TTH(MN)*CS                                                  26460
      AQS=AQS+QS(MN)*CS                                                     26470
      ANXTH=ANXTH+TXT(MN)*SN                                                26480
   72 PF(K)=PF(K)+PHI(MN)*SN                                                26490
  429 CONTINUE                                                              26500
      IF(K.EQ.1) WRITE(6,117)                                               26510
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))           26520
     1 GO TO 220                                                            26530
      GO TO 221                                                             26540
  220 WRITE (6,1181) K,ANX,ANTH,ANXTH,AMX,AMTH,AMXTH                        26550
      GO TO 1354                                                            26560
  221 WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH                      26570
 1354 IF ($MODAL.OR.(ICHCK1.EQ.0)) GO TO 2                                  26580
      YNS(K)=ANX                                                            26590
      YNTH(K)=ANTH                                                          26600
      YNSTH(K)=ANXTH                                                        26610
      YQS(K)=AQS                                                            26620
      YMS(K)=AMX                                                            26630
      YMTH(K)=AMTH                                                          26640
      YMSTH(K)=AMXTH                                                        26650
    2 CONTINUE                                                              26660
  121 CONTINUE                                                              26670
      DO 66 K=1,KMAX                                                        26680
      FK=K-1                                                                26690
      FIFREQ=IFREQ                                                          26700
      KTST=(K-1)/IFREQ                                                      26710
      FKTST=KTST                                                            26720
      FKTEST=FK/FIFREQ-FKTST                                                26730
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 661                                     26740
      IF(FKTEST.NE.0.) GO TO 658                                            26750
  661 IF(K.EQ.1) WRITE(6,217)                                               26760
      WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),PTF(K),PF(K)               26770
      IF ($MODAL.OR.(ICHCK2.EQ.0)) GO TO 658                                26780
      YU(K)=X(1,K)                                                          26790
      YV(K)=X(2,K)                                                          26800
      YW(K)=X(3,K)                                                          26810
      YPHIS(K)=X(4,K)                                                       26820
      YPHIT(K)=PTF(K)                                                       26830
      YPHI(K)=PF(K)                                                         26840
```

```
658 DO 659 I=1,4                                                                26850
659 X(I,K)=0.                                                                   26860
660 CONTINUE                                                                    26870
    IF ($PLOTS.AND..NOT.$MODAL.AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))            26880
   1   CALL PLOT2(NTH)                                                          26890
 21 CONTINUE                                                                    26900
991 IF(IMODE.LE.0) RETURN                                                       26910
    DO 534 MN=1,MNMAXO                                                          26920
    IF(.NOT.$VIBES) WRITE(6,749) N(MN)                                          26930
    DO 521 MM=1,MNMAXG                                                          26940
    I1=1+(MM-1)*KMAX2                                                           26950
    I2=I1+1                                                                     26960
    U1(MM)=Z(1,I1)                                                              26970
    U2(MM)=Z(1,I2)                                                              26980
    V1(MM)=Z(2,I1)                                                              26990
    V2(MM)=Z(2,I2)                                                              27000
    W1(MM)=Z(3,I1)                                                              27010
    W2(MM)=Z(3,I2)                                                              27020
521 CONTINUE                                                                    27030
    DO 445 K=1, KMAX                                                            27040
    K1=K+1                                                                      27050
    CALL BDB(K,BS,DB,DS,DD)                                                     27060
    IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)                                     27070
    IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)                                  27080
    TXZ=TX(MN)                                                                  27090
    TTHZ=TTH(MN)                                                                27100
    TXTZ=TXT(MN)                                                                27110
    AMXZ=MX(MN)                                                                 27120
    AMTHZ=MTH(MN)                                                               27130
    AMXTZ=MXT(MN)                                                               27140
    QSZ=QS(MN)                                                                  27150
    X(1,K)=PHIX(MN)                                                             27160
    X(2,K)=PHIT(MN)                                                             27170
    X(3,K)=PHI(MN)                                                              27180
    IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 583                                        27190
    IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 583                                     27200
    CALL PHIBET(K)                                                              27210
    DEX=DEGMX(K)                                                                27220
    RRA=1./R(K)                                                                 27230
    QX=QMXI(K)                                                                  27240
    QT=QMT(K)                                                                   27250
    GA=GAM(K)                                                                   27260
    DQXT=QX-QT                                                                  27270
    GDQ=GA*DQXT                                                                 27280
    DD2D=DD2*DS                                                                 27290
    EN=N(MN)                                                                    27300
    ENR=EN*RRA                                                                  27310
    CALL TLOAD(K)                                                               27320
```

```
      TTS=TT(MN)*ALOAD                                                          27330
      EX=(U3(MN)-U1(MN))*TDLI+OX*W2(MN)+ENL*OSE*(BX3(MN)+BE3(MN))               27340
      ET=ENR*V2(MN)+GA*U2(MN)+OT*W2(MN)+ENL*OSE*(BT3(MN)+BE3(MN))               27350
      EXT=.5*((V3(MN)-V1(MN))*TDLI-ENR*U2(MN)-GA*V2(MN)+ENL*SOE*BXT3(MN)        27360
     1)                                                                         27370
      KT=ENR*PHIT(MN)+GA*PHIX(MN)                                               27380
      KXT=.5*(ENR*(-PHIX(MN)-GA*W2(MN)+(W3(MN)-W1(MN))*TDLI)+GDO*V2(MN)         27390
     1     +OT*(V3(MN)-V1(MN))*TDLI-GA*PHIT(MN)-DCXT*PHI(MN))                   27400
      TXZ=(BS*(EX+NU*ET)-TTS)*ABZ                                               27410
      TTHZ=(BS*(ET+NU*EX)-TTS)*ABZ                                              27420
      TXTZ=BS*D1*EXT*ABZ                                                        27430
      MK1=K1+(MN-1)*KMAX2                                                       27440
      AMXZ=Z(4,MK1)                                                             27450
      AMTHZ=NU*AMXZ+DD2D*KT-D1*MT(MN)*ALOAD                                     27460
      AMXTZ=DS*D1*KXT                                                           27470
      MK11=MK1+1                                                                27480
      MKK1=MK1-1                                                                27490
      QSZ=SIGO*TKN*LAM2*(GA*AMXZ+(Z(4,MK11)-Z(4,MKK1))*TDLI+ENR*AMXTZ           27500
     1     -GA*AMTHZ)                                                           27510
      AMXZ=AMXZ*ABZ3                                                            27520
      AMTHZ=AMTHZ*ABZ3                                                          27530
      AMXTZ=AMXTZ*ABZ3                                                          27540
      X(1,K)=PHIX(MN)*ABZO                                                      27550
      X(2,K)=PHIT(MN)*ABZO                                                      27560
      X(3,K)=PHI(MN)*ABZO                                                       27570
      DO 533 MM=1,MNMAXO                                                        27580
      U1(MM)=U2(MM)                                                             27590
      U2(MM)=U3(MM)                                                             27600
      V1(MM)=V2(MM)                                                             27610
      V2(MM)=V3(MM)                                                             27620
      W1(MM)=W2(MM)                                                             27630
  533 W2(MM)=W3(MM)                                                             27640
      FK=K-1                                                                    27650
      FIFREQ=IFREQ                                                              27660
      KTST=(K-1)/IFREQ                                                          27670
      FKTST=KTST                                                                27680
      FKTEST=FK/FIFREQ-FKTST                                                    27690
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583                                         27700
      IF(FKTEST.NE.0.) GO TO 445                                                27710
  583 CONTINUE                                                                  27720
      IF(K.EQ.1) WRITE(6,117)                                                   27730
      IF((K.EQ.1.AND.IBCINL.LT.0).OR.(K.EQ.KMAX.AND.IBCFNL.LT.0))               27740
     1 GO TO 4444                                                               27750
      GO TO 4445                                                                27760
 4444 WRITE (6,1181) K,TXZ,TTHZ,TXTZ,AMXZ,AMTHZ,AMXTZ                          27770
      GO TO 4446                                                                27780
 4445 WRITE(6,118) K,TXZ,TTHZ,TXTZ,QSZ,AMXZ,AMTHZ,AMXTZ                        27790
 4446 IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK1.EQ.0)) GO TO 445                27800
```

```
      YNS(K)=TXZ                                                        27810
      YNTH(K)=TTHZ                                                      27820
      YNSTH(K)=TXTZ                                                     27830
      YQS(K)=QSZ                                                        27840
      YMS(K)=AMXZ                                                       27850
      YMTH(K)=AMTHZ                                                     27860
      YMSTH(K)=AMXTZ                                                    27870
  445 CONTINUE                                                          27880
  446 WRITE(6,217)                                                      27890
      DO 447 K=1,KMAX                                                   27900
      FK=K-1                                                            27910
      FIFREQ=IFREQ                                                      27920
      KTST=(K-1)/IFREQ                                                  27930
      FKTST=KTST                                                        27940
      FKTEST=FK/FIFREQ-FKTST                                            27950
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 593                                 27960
      IF(FKTEST.NE.0.) GO TO 447                                        27970
  593 KZ=K+1+(MN-1)*KMAX2                                               27980
      UP=Z(1,KZ)*ABZN                                                   27990
      VP=Z(2,KZ)*ABZN                                                   28000
      WP=Z(3,KZ)*ABZN                                                   28010
      WRITE(6,218) K,UP,VP,WP,X(1,K),X(2,K),X(3,K)                      28020
      IF (.NOT.$PLOTS.OR..NOT.$MODAL.OR.(ICHCK2.EQ.0)) GO TO 447        28030
      YU(K)=UP                                                          28040
      YV(K)=VP                                                          28050
      YW(K)=WP                                                          28060
      YPHIS(K)=X(1,K)                                                   28070
      YPHIT(K)=X(2,K)                                                   28080
      YPHI(K)=X(3,K)                                                    28090
  447 CONTINUE                                                          28100
      IF ($PLOTS.AND.$MODAL.AND.((ICHCK1.GT.0).OR.(ICHCK2.GT.0)))       28110
     1   CALL PLOT2(1)                                                  28120
  534 CONTINUE                                                          28130
C*********************************************************************  28140
  101 FORMAT('1',' THE LOAD STEP NUMBER IS ',I2,' THE L                 28150
     1OAD FACTOR IS ',E11.4,' THE SOLUTION CONVERGED IN ',I2,'           28160
     2ITERATIONS'////)                                                  28170
  116 FORMAT('0',' THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AN         28180
     1D ROTATIONS FOLLOW FOR THETA =',E15.6///)                         28190
  117 FORMAT (/' STATION  N S  N THETA  N STHETA                        28200
     1  Q S  M S  M THETA  M STHETA'//)                                 28210
  118 FORMAT(1X,I3,3X,7E16.4)                                           28220
 1181 FORMAT(1X,I3,3X,3E16.4,' NOT COMPUTED ',3E16.4)                   28230
  151 FORMAT('1',' THE TIME STEP NUMBER IS ',I4,' THE TIME IS ',F5       28240
     1.2,' OR ',F9.3,' SECONDS THE SOLUTION CONVERGED IN ',I2,' ITE     28250
     3RATIONS'////)                                                     28260
  217 FORMAT(//' STATION  U  V  W                                       28270
     1  PHI S  PHI THETA  PHI'/)                                        28280
```

```
  218 FORMAT(1X,I3,3X,6E16.4)                                                    28290
  749 FORMAT('1',40X,' MODAL OUTPUT FOR MODE N = ',I3,' FOLLOWS')                28300
C*****************************************************************************28310
      RETURN                                                                     28320
      END                                                                        28330

      SUBROUTINE PLOT1(I)                                                        28340
C*****************************************************************************28350
C     THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER     *       28360
C     SPECIFIED) INPUT QUANTITIES                                       *       28370
C*****************************************************************************28380
      IMPLICIT LOGICAL*1 ($)                                                     28390
      COMMON /IBL4/ KMAX,KL                                                      28400
      COMMON /BLPLOT/ IRADII,IGAMMA,IOMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,          28410
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,           28420
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,              28430
     3                IPHIT,IPHI,$PLOTS,$MODAL                                   28440
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),           28450
     1                YDEOMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),           28460
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),           28470
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),           28480
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),         28490
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),             28500
     6                YPHI(100),XSTATN(100)                                      28510
C*****************************************************************************28520
      NGKMAX=-KMAX                                                               28530
      IF (I.GT.1) GO TO 1                                                        28540
      IF (IRADII.EQ.0) GO TO 2                                                   28550
      WRITE (6,1000)                                                             28560
      CALL PLOTIT (XSTATN,XRADII,NGKMAX,0)                                       28570
      WRITE (6,1001)                                                             28580
    2 IF (IGAMMA.EQ.0) GO TO 4                                                   28590
      WRITE (6,1000)                                                             28600
      CALL PLOTIT (XSTATN,YGAMMA,NGKMAX,0)                                       28610
      WRITE (6,1002)                                                             28620
    4 IF (IOMEGS.EQ.0) GO TO 5                                                   28630
      WRITE (6,1000)                                                             28640
      CALL PLOTIT (XSTATN,YOMEGS,NGKMAX,0)                                       28650
      WRITE (6,1003)                                                             28660
    5 IF (IOMEGT.EQ.0) GO TO 6                                                   28670
      WRITE (6,1000)                                                             28680
      CALL PLOTIT (XSTATN,YOMEGT,NGKMAX,0)                                       28690
      WRITE (6,1004)                                                             28700
    6 IF (IDEOMS.EQ.0) GO TO 7                                                   28710
      WRITE (6,1000)                                                             28720
      CALL PLOTIT (XSTATN,YDEOMS,NGKMAX,0)                                       28730
      WRITE (6,1005)                                                             28740
```

```
    7 IF (IBSTIF.EQ.0) GO TO 8                                          28750
      WRITE (6,1000)                                                    28760
      CALL PLOTIT (XSTATN,YBSTIF,NGKMAX,0)                              28770
      WRITE (6,1006)                                                    28780
    8 IF (IDSTIF.EQ.0) GO TO 9                                          28790
      WRITE (6,1000)                                                    28800
      CALL PLOTIT (XSTATN,YDSTIF,NGKMAX,0)                              28810
      WRITE (6,1007)                                                    28820
    9 IF (IBBSTF.EQ.0) GO TO 10                                         28830
      WRITE (6,1000)                                                    28840
      CALL PLOTIT (XSTATN,YBBSTF,NGKMAX,0)                              28850
      WRITE (6,1008)                                                    28860
   10 IF (IDDSTF.EQ.0) GO TO 1                                          28870
      WRITE (6,1000)                                                    28880
      CALL PLOTIT (XSTATN,YDDSTF,NGKMAX,0)                              28890
      WRITE (6,1009)                                                    28900
    1 IF (IPR.EQ.0) GO TO 11                                            28910
      WRITE (6,1000)                                                    28920
      CALL PLOTIT (XSTATN,YPR,NGKMAX,0)                                 28930
      WRITE (6,1010)                                                    28940
   11 IF (IPS.EQ.0) GO TO 12                                            28950
      WRITE (6,1000)                                                    28960
      CALL PLOTIT (XSTATN,YPS,NGKMAX,0)                                 28970
      WRITE (6,1011)                                                    28980
   12 IF (IPT.EQ.0) GO TO 13                                            28990
      WRITE (6,1000)                                                    29000
      CALL PLOTIT (XSTATN,YPT,NGKMAX,0)                                 29010
      WRITE (6,1012)                                                    29020
   13 IF (ITT.EQ.0) GO TO 14                                            29030
      WRITE (6,1000)                                                    29040
      CALL PLOTIT (XSTATN,YTT,NGKMAX,0)                                 29050
      WRITE (6,1013)                                                    29060
   14 IF (IMT.EQ.0) GO TO 15                                            29070
      WRITE (6,1000)                                                    29080
      CALL PLOTIT (XSTATN,YMT,NGKMAX,0)                                 29090
      WRITE (6,1014)                                                    29100
   15 IF (IDTT.EQ.0) GO TO 16                                           29110
      WRITE (6,1000)                                                    29120
      CALL PLOTIT (XSTATN,YDTT,NGKMAX,0)                                29130
      WRITE (6,1015)                                                    29140
   16 IF (IDMT.EQ.0) GO TO 17                                           29150
      WRITE (6,1000)                                                    29160
      CALL PLOTIT (XSTATN,YDMT,NGKMAX,0)                                29170
      WRITE (6,1016)                                                    29180
      WRITE (6,1000)                                                    29190
   17 RETURN                                                            29200
C*****************************************************************     29210
 1000 FORMAT ('1')                                                      29220
```

```
 1001 FORMAT ('0',T10,'RADIUS VS STATION')                                  29230
 1002 FORMAT ('0',T10,'GAMMA VS STATION')                                   29240
 1003 FORMAT ('0',T10,'OMEGA-S VS STATION')                                 29250
 1004 FORMAT ('0',T10,'OMEGA-THETA VS STATION')                             29260
 1005 FORMAT ('0',T10,'DECMEGA-S VS STATION')                               29270
 1006 FORMAT ('0',T10,'B-STIFFNESS VS STATION')                             29280
 1007 FORMAT ('0',T10,'D-STIFFNESS VS STATION')                             29290
 1008 FORMAT ('0',T10,'DB-STIFFNESS VS STATION')                            29300
 1009 FORMAT ('0',T10,'DD-STIFFNESS VS STATION')                            29310
 1010 FORMAT ('0',T10,'NORMAL LOADING VS STATION')                          29320
 1011 FORMAT ('0',T10,'MERIDIO-TANGENTIAL LOADING VS STATION')              29330
 1012 FORMAT ('0',T10,'CIRCUMFERENTIO-TANGENTIAL LOADING VS STATION')       29340
 1013 FORMAT ('0',T10,'THERMAL LOADING VS STATION')                         29350
 1014 FORMAT ('0',T10,'THERMAL BENDING LOADING VS STATION')                 29360
 1015 FORMAT ('0',T10,'DE-THERMAL LOADING VS STATION')                      29370
 1016 FORMAT ('0',T10,'DE-THERMAL-BENDING LOADING VS STATION')              29380
C*******************************************************************      29390
      END                                                                   29400

      SUBROUTINE PLOT2(NTH)                                                 29410
C*******************************************************************      29420
C     THIS SUBROUTINE CALLS PLOTTING ROUTINES FOR APPROPRIATE (USER    *    29430
C     SPECIFIED) OUTPUT QUANTITIES                                     *    29440
C*******************************************************************      29450
      IMPLICIT LOGICAL*1 ($)                                                29460
      COMMON /IBL4/ KMAX,KL                                                 29470
      COMMON /BL19/ TH(36)                                                  29480
      COMMON /BLPLOT/ IRADII,IGAMMA,ICMEGS,IOMEGT,IDEOMS,IBSTIF,IDSTIF,     29490
     1                IBBSTF,IDDSTF,IPR,IPS,IPT,ITT,IMT,IDTT,IDMT,INS,      29500
     2                INTH,INSTH,IQS,IMS,IMTH,IMSTH,IU,IV,IW,IPHIS,         29510
     3                IPHIT,IPHI,$PLOTS,$MODAL                              29520
      COMMON /BLPLT1/ XRADII(100),YGAMMA(100),YOMEGS(100),YOMEGT(100),      29530
     1                YDECMS(100),YBSTIF(100),YDSTIF(100),YBBSTF(100),      29540
     2                YDDSTF(100),YPR(100),YPS(100),YPT(100),YTT(100),      29550
     3                YMT(100),YDTT(100),YDMT(100),YNS(100),YNTH(100),      29560
     4                YNSTH(100),YQS(100),YMS(100),YMTH(100),YMSTH(100),    29570
     5                YU(100),YV(100),YW(100),YPHIS(100),YPHIT(100),        29580
     6                YPHI(100),XSTATN(100)                                 29590
C*******************************************************************      29600
      NGKMAX=-KMAX                                                          29610
      IF ($MODAL) GO TO 121                                                 29620
      IF (INS.EQ.0) GO TO 4                                                 29630
      WRITE (6,1000)                                                        29640
      IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)                         29650
      IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)                       29660
      WRITE (6,1001)TH(NTH)                                                 29670
    4 IF (INTH.EQ.0) GO TO 5                                                29680
```

```
      WRITE (6,1000)                                                    29690
      IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)                   29700
      IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)                 29710
      WRITE (6,1002) TH(NTH)                                            29720
    5 IF (INSTH.EQ.0) GO TO 6                                           29730
      WRITE (6,1000)                                                    29740
      IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)                 29750
      IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)               29760
      WRITE (6,1003) TH(NTH)                                            29770
    6 IF (IQS.EQ.0) GO TO 7                                             29780
      WRITE (6,1000)                                                    29790
      IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)                     29800
      IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)                   29810
      WRITE (6,1004) TH(NTH)                                            29820
    7 IF (IMS.EQ.0) GO TO 8                                             29830
      WRITE (6,1000)                                                    29840
      IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)                     29850
      IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)                   29860
      WRITE (6,1005) TH(NTH)                                            29870
    8 IF (IMTH.EQ.0) GO TO 9                                            29880
      WRITE (6,1000)                                                    29890
      IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)                   29900
      IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)                 29910
      WRITE (6,1006) TH(NTH)                                            29920
    9 IF (IMSTH.EQ.0) GO TO 1211                                        29930
      WRITE (6,1000)                                                    29940
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                 29950
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)               29960
      WRITE (6,1007) TH(NTH)                                            29970
 1211 IF (IU.EQ.0) GO TO 10                                             29980
      WRITE (6,1000)                                                    29990
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                       30000
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                     30010
      WRITE (6,1010) TH(NTH)                                            30020
   10 IF (IV.EQ.0) GO TO 11                                             30030
      WRITE (6,1000)                                                    30040
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                       30050
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                     30060
      WRITE (6,1009) TH(NTH)                                            30070
   11 IF (IW.EQ.0) GO TO 12                                             30080
      WRITE (6,1000)                                                    30090
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                       30100
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                     30110
      WRITE (6,1008) TH(NTH)                                            30120
   12 IF (IPHIS.EQ.0) GO TO 13                                          30130
      WRITE (6,1000)                                                    30140
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                 30150
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)               30160
```

```
      WRITE (6,1011) TH(NTH)                                                    30170
   13 IF (IPHIT.EQ.0) GO TO 14                                                  30180
      WRITE (6,1000)                                                            30190
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                         30200
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)                       30210
      WRITE (6,1012) TH(NTH)                                                    30220
   14 IF (IPHI.EQ.0) GO TO 21                                                   30230
      WRITE (6,1000)                                                            30240
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)                         30250
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)                       30260
      WRITE (6,1013) TH(NTH)                                                    30270
   21 RETURN                                                                    30280
  121 IF (INS.EQ.0) GO TO 15                                                    30290
      WRITE (6,1000)                                                            30300
      IF (INS.GT.0) CALL PLOTIT (XSTATN,YNS,KMAX,0)                             30310
      IF (INS.LT.0) CALL PLOTIT (XSTATN,YNS,NGKMAX,0)                           30320
      WRITE (6,2001)                                                            30330
   15 IF (INTH.EQ.0) GO TO 16                                                   30340
      WRITE (6,1000)                                                            30350
      IF (INTH.GT.0) CALL PLOTIT (XSTATN,YNTH,KMAX,0)                           30360
      IF (INTH.LT.0) CALL PLOTIT (XSTATN,YNTH,NGKMAX,0)                         30370
      WRITE (6,2002)                                                            30380
   16 IF (INSTH.EQ.0) GO TO 17                                                  30390
      WRITE (6,1000)                                                            30400
      IF (INSTH.GT.0) CALL PLOTIT (XSTATN,YNSTH,KMAX,0)                         30410
      IF (INSTH.LT.0) CALL PLOTIT (XSTATN,YNSTH,NGKMAX,0)                       30420
      WRITE (6,2003)                                                            30430
   17 IF (IQS.EQ.0) GO TO 18                                                    30440
      WRITE (6,1000)                                                            30450
      IF (IQS.GT.0) CALL PLOTIT (XSTATN,YQS,KMAX,0)                             30460
      IF (IQS.LT.0) CALL PLOTIT (XSTATN,YQS,NGKMAX,0)                           30470
      WRITE (6,2004)                                                            30480
   18 IF (IMS.EQ.0) GO TO 19                                                    30490
      WRITE (6,1000)                                                            30500
      IF (IMS.GT.0) CALL PLOTIT (XSTATN,YMS,KMAX,0)                             30510
      IF (IMS.LT.0) CALL PLOTIT (XSTATN,YMS,NGKMAX,0)                           30520
      WRITE (6,2005)                                                            30530
   19 IF (IMTH.EQ.0) GO TO 22                                                   30540
      WRITE (6,1000)                                                            30550
      IF (IMTH.GT.0) CALL PLOTIT (XSTATN,YMTH,KMAX,0)                           30560
      IF (IMTH.LT.0) CALL PLOTIT (XSTATN,YMTH,NGKMAX,0)                         30570
      WRITE (6,2006)                                                            30580
   22 IF (IMSTH.EQ.0) GO TO 231                                                 30590
      WRITE (6,1000)                                                            30600
      IF (IMSTH.GT.0) CALL PLOTIT (XSTATN,YMSTH,KMAX,0)                         30610
      IF (IMSTH.LT.0) CALL PLOTIT (XSTATN,YMSTH,NGKMAX,0)                       30620
      WRITE (6,2007)                                                            30630
  231 IF (IU.EQ.0) GO TO 23                                                     30640
```

```
      WRITE (6,1000)                                                    30650
      IF (IU.GT.0) CALL PLOTIT (XSTATN,YU,KMAX,0)                       30660
      IF (IU.LT.0) CALL PLOTIT (XSTATN,YU,NGKMAX,0)                     30670
      WRITE (6,2010)                                                    30680
   23 IF (IV.EQ.0) GO TO 24                                             30690
      WRITE (6,1000)                                                    30700
      IF (IV.GT.0) CALL PLOTIT (XSTATN,YV,KMAX,0)                       30710
      IF (IV.LT.0) CALL PLOTIT (XSTATN,YV,NGKMAX,0)                     30720
      WRITE (6,2009)                                                    30730
   24 IF (IW.EQ.0) GO TO 25                                             30740
      WRITE (6,1000)                                                    30750
      IF (IW.GT.0) CALL PLOTIT (XSTATN,YW,KMAX,0)                       30760
      IF (IW.LT.0) CALL PLOTIT (XSTATN,YW,NGKMAX,0)                     30770
      WRITE (6,2008)                                                    30780
   25 IF (IPHIS.EQ.0) GO TO 26                                          30790
      WRITE (6,1000)                                                    30800
      IF (IPHIS.GT.0) CALL PLOTIT (XSTATN,YPHIS,KMAX,0)                 30810
      IF (IPHIS.LT.0) CALL PLOTIT (XSTATN,YPHIS,NGKMAX,0)               30820
      WRITE (6,2011)                                                    30830
   26 IF (IPHIT.EQ.0) GO TO 27                                          30840
      WRITE (6,1000)                                                    30850
      IF (IPHIT.GT.0) CALL PLOTIT (XSTATN,YPHIT,KMAX,0)                 30860
      IF (IPHIT.LT.0) CALL PLOTIT (XSTATN,YPHIT,NGKMAX,0)               30870
      WRITE (6,2012)                                                    30880
   27 IF (IPHI.EQ.0) GO TO 28                                           30890
      WRITE (6,1000)                                                    30900
      IF (IPHI .GT.0) CALL PLOTIT (XSTATN,YPHI ,KMAX,0)                 30910
      IF (IPHI .LT.0) CALL PLOTIT (XSTATN,YPHI ,NGKMAX,0)               30920
      WRITE (6,2013)                                                    30930
   28 RETURN                                                            30940
C*****************************************************************     30950
 1000 FORMAT ('1')                                                      30960
 1001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THETA = 30970
     1 ',F10.5,'  RADIANS')                                             30980
 1002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATN,  MERIDIAN AT THE 30990
     1TA = ',F10.5,'  RADIANS')                                         31000
 1003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATN,  MERIDIAN AT T 31010
     1HETA = ',F10.5,'  RADIANS')                                       31020
 1004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATN,  MERIDIAN AT THETA = ', 31030
     1F10.5,'  RADIANS')                                               31040
 1005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATN,  MERIDIAN AT THETA = 31050
     1 ',F10.5,' RADIANS')                                              31060
 1006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATN,  MERIDIAN AT THETA 31070
     1 = ',F10.5,'  RADIANS')                                           31080
 1007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATN,  MERIDIAN AT THE 31090
     1TA = ',F10.5,'  RADIANS')                                         31100
 1008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATN,  MERIDIAN AT THETA =  31110
     1',F10.5,'  RADIANS')                                             31120
```

```
 1009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATN,  MERIDIAN AT   31130
     1 THETA = ',F10.5,'  RADIANS')                                        31140
 1010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATN,  MERIDIAN AT THET   31150
     1A = ',F10.5,'  RADIANS')                                             31160
 1011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATN,  MERIDIAN AT THETA    31170
     1= ',F10.5,'  RADIANS')                                               31180
 1012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATN,  MERIDIAN AT T   31190
     1HETA = ',F10.5,'  RADIANS')                                          31200
 1013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATN,  MERID   31210
     1IAN AT THETA = ',F10.5,'  RADIANS')                                  31220
 2001 FORMAT ('0',T10,'"S"-MEMBRANE FORCE VS STATION')                     31230
 2002 FORMAT ('0',T10,'"THETA"-MEMBRANE FORCE VS STATION')                 31240
 2003 FORMAT ('0',T10,'"S-THETA" MEMBRANE FORCE VS STATION')               31250
 2004 FORMAT ('0',T10,'EFFECTIVE SHEAR VS STATION')                        31260
 2005 FORMAT ('0',T10,'"S"-BENDING MOMENT VS STATION')                     31270
 2006 FORMAT ('0',T10,'THETA-BENDING MOMENT VS STATION')                   31280
 2007 FORMAT ('0',T10,'S-THETA BENDING MOMENT VS STATION')                 31290
 2008 FORMAT ('0',T10,'NORMAL DEFLECTION VS STATION')                      31300
 2009 FORMAT ('0',T10,'CIRCUMFERENTIAL DEFLECTION VS STATION')             31310
 2010 FORMAT ('0',T10,'MERIDIONAL DEFLECTION VS STATION')                  31320
 2011 FORMAT ('0',T10,'MERIDIONAL ROTATION VS STATION')                    31330
 2012 FORMAT ('0',T10,'CIRCUMFERENTIAL ROTATION VS STATION')               31340
 2013 FORMAT ('0',T10,'MERIDIO-CIRCUMFERENTIAL ROTATION VS STATION')       31350
C*************************************************************************31360
      END                                                                  31370


      SUBROUTINE PLOTIT(X,Y,NN,MODCUR)                                     31380
C*************************************************************************31390
C     THIS SUBROUTINE AND THE THREE THAT FOLLOW IT COMPRISE THE SELF-   *  31400
C CONTAINED PLOTTING CAPABILITY OF THE PROGRAM, SATANS.  THEY RECEIVE   *  31410
C THE DATA TO BE PLOTTED, ROUND IT, SCALE IT AND DRAW IT ON THE HIGH-   *  31420
C SPEED LINE PRINTER.                                                   *  31430
C*************************************************************************31440
      DIMENSION X( 1), Y( 1), RANGES(4)                                    31450
      EQUIVALENCE (RANGES(1),XMAX), (RANGES(2),XMIN), (RANGES(3),YMAX),    31460
     1           (RANGES(4),YMIN)                                          31470
      KN=IABS(NN)                                                          31480
      IF(MODCUR.GT.1) GO TO 5                                              31490
C*************************************************************************31500
C     FIND MAX & MIN FOR SCALE COMPUTATIONS                             *  31510
C*************************************************************************31520
      XMAX=-1.E20                                                          31530
      XMIN=1.E20                                                           31540
      YMAX=-1.E20                                                          31550
      YMIN=1.E20                                                           31560
      DO 1 I=1,KN                                                          31570
      IF(X(I).LT.XMAX) GO TO 6                                             31580
```

```
      XMAX=X(I)                                                          31590
    6 IF(X(I).GT.XMIN) GO TO 7                                           31600
      XMIN=X(I)                                                          31610
    7 IF(Y(I).LT.YMAX) GO TO 8                                           31620
      YMAX=Y(I)                                                          31630
    8 IF(Y(I).GT.YMIN) GO TO 1                                           31640
      YMIN=Y(I)                                                          31650
    1 CONTINUE                                                           31660
C*****************************************************************     31670
C     IF NOT AUTOSCALE GO TO CALL DRAWIT                              *  31680
C*****************************************************************     31690
      IF(NN.GT.0) GO TO 5                                                31700
C*****************************************************************     31710
C     COMPUTE X-SCALE & NEW XMAX AND XMIN                             *  31720
C*****************************************************************     31730
      CALL SCALIT(XMAX,XMIN,4)                                           31740
C*****************************************************************     31750
C     COMPUTE Y-SCALE & NEW YMAX AND YMIN                             *  31760
C*****************************************************************     31770
      CALL SCALIT(YMAX,YMIN,6)                                           31780
C*****************************************************************     31790
C     PLOT CURVE                                                      *  31800
C*****************************************************************     31810
    5 CALL DRAWIT(X,Y,KN,RANGES,1,MODCUR)                                31820
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                              31830
C*****************************************************************     31840
C     PRINT SCALES WHEN LAST CURVE PLOTTED                            *  31850
C*****************************************************************     31860
      XS=(XMAX-XMIN)/80.                                                 31870
      YS=(YMAX-YMIN)/60.                                                 31880
      WRITE(6,100) XS,YS                                                 31890
  100 FORMAT(  15X,'X-SCALE:  "*"=',E10.3,' UNITS'//                     31900
     1         15X,'Y-SCALE:  "*"=',E10.3,' UNITS')                      31910
      RETURN                                                             31920
C*****************************************************************     31930
      END                                                                31940

      SUBROUTINE SCALIT(XMAX,XMIN,IDIV)                                  31950
C*****************************************************************     31960
C     ROUND MAXIMUM TO NEXT HIGHEST 2 SIG FIGS                        *  31970
C*****************************************************************     31980
      DIV=IDIV                                                           31990
      XMAX=AMAX1(0.,XMAX)                                                32000
      CALL ROUND(XMAX,IMX,FMX)                                           32010
      IMX=IMX-1                                                          32020
      FMX=FMX*10.                                                        32030
    3 XMX=FMX*10.**IMX                                                   32040
```

```
      IF(XMX.GE.XMAX) GO TO 2                                                   32050
      FMX=FMX+1.                                                                32060
      IMM=FMX                                                                   32070
      FMX=IMM                                                                   32080
      GO TO 3                                                                   32090
C*************************************************************************    32100
C     ROUND MINIMUM TO NEXT LOWEST 2 SIG FIGS                             *     32110
C*************************************************************************    32120
    2 XMIN=AMIN1(0.,XMIN)                                                       32130
      CALL ROUND(XMIN,IMN,FMN)                                                  32140
      IMN=IMN-1                                                                 32150
      FMN=FMN*10.                                                               32160
   14 XMN=FMN*10.**IMN                                                          32170
      IF(XMIN.GE.XMN) GO TO 11                                                  32180
      FMN=FMN-1.                                                                32190
      IMM=FMN                                                                   32200
      FMN=IMM                                                                   32210
      GO TO 14                                                                  32220
C*************************************************************************    32230
C     ROUND MAX & MIN TO 1. OR .1 IF RANGE LARGE                          *     32240
C*************************************************************************    32250
   11 XSC=XMX-XMN                                                               32260
      IM=0                                                                      32270
      SM=1.                                                                     32280
    9 IF(XSC/DIV.LE.SM) GO TO 12                                                32290
      IF(ABS(XMN).LT.SM.AND.ABS(XMN).GT.0.) XMN=SIGN(SM,XMN)                    32300
      IF(ABS(XMX).LT.SM.AND.ABS(XMX).GT.0.) XMX=SIGN(SM,XMX)                    32310
   12 IF(IM.GT.0) GO TO 19                                                      32320
      SM=.1                                                                     32330
      IM=IM+1                                                                   32340
      GO TO 9                                                                   32350
C*************************************************************************    32360
C     ROUND RANGE (MAX-MIN) TO 2 SIG FIGS                                 *     32370
C*************************************************************************    32380
   19 XSC=XMX-XMN                                                               32390
      CALL ROUND(XSC,IS10,FACTX)                                                32400
C*************************************************************************    32410
C     FIND FACTOR WHICH IS MULTIPLE OF IDIV                               *     32420
C*************************************************************************    32430
      FACTX=FACTX*10.                                                           32440
      OFAC=FACTX                                                                32450
      IS10=IS10-1                                                               32460
      IFX=FACTX                                                                 32470
      FACTX=IFX                                                                 32480
   20 IF(MOD(IFX,IDIV).EQ.0.AND.FACTX.GE.OFAC) GO TO 10                         32490
      IFX=IFX+1                                                                 32500
      FACTX=IFX                                                                 32510
      GO TO 20                                                                  32520
```

```
   10 IF(IDIV.GT.4) GO TO 15
C*****************************************************************
C     IF X SCALE BETWEEN 8. AND 10., ROUND TO 10.                *
C*****************************************************************
      FFX=ABS(FACTX/10.)
      IF(FFX.GT.8..AND.FFX.LT.10.) FFX=10.
      IF(FACT X.LT.0.) FXX=-10.
      FACTX=FFX*10.
   15 XSC=FACTX*10.**IS10
C*****************************************************************
C     COMPUTE NEW MAX & MIN FROM ROUNDED SCALE                   *
C*****************************************************************
      IF(XM N*XM X.NE.0.) GO TO 4
      IF(XM N.LT.0.) XMIN=-XSC
      IF(XM X.GT.0.) XMAX=XSC
      RETURN
    4 XMAX=XSC+XMN
      XMIN=XMN
      RETURN
C*****************************************************************
      END


      SUBROUTINE ROUND(ANUM,IS,FACT)
C*****************************************************************
C     EXPRESS ANUM IN SCIENTIFIC NOTATION WHERE                  *
C           ANUM=FACT*10.**IS WHERE FACT IS BETWEEN 1. AND 9.9   *
C*****************************************************************
      IF(ANUM.EQ.0.) GO TO 15
      BNUM=ANUM
      IF(BNUM.LT.0.) BNUM=-BNUM
      IS=ALOG(BNUM)*.43429448
      FACT=BNUM/10.**IS
C*****************************************************************
C     FIND POWER OF 10                                           *
C*****************************************************************
      IDD=-3
      R2=0
      DO 10 II=1,5
      IDD=IDD+1
      R1=R2
      R2=10.**(IDD+1)
      IF(FACT.GE.R1.AND.FACT.LT.R2) GO TO 8
   10 CONTINUE
    8 FACT=FACT*10.**(-IDD)
      IS=IS+IDD
C*****************************************************************
C     ROUND MANTISSA TO 2 SIG FIGS                               *
```

```
C*************************************************************************      32990
      IFAC=FACT*10.+.05                                                        33000
      FACT=IFAC                                                                33010
      FACT=FACT/10.                                                            33020
      IF(FACT.LT.10.) GO TO 20                                                 33030
C*************************************************************************      33040
C     SET TO 1 IF LESS THAN 10.                                          *     33050
C*************************************************************************      33060
      FACT=1.                                                                  33070
      IS=IS+1                                                                  33080
C*************************************************************************      33090
C     IF INPUT NEGATIVE, SET MANTISSA NEGATIVE                           *     33100
C*************************************************************************      33110
   20 IF(ANUM.LT.0.) FACT=-FACT                                                33120
      RETURN                                                                   33130
C*************************************************************************      33140
C     SET TO 0. IF 0.                                                    *     33150
C*************************************************************************      33160
   15 FACT=0.                                                                  33170
      IS=0                                                                     33180
      RETURN                                                                   33190
C*************************************************************************      33200
      END                                                                      33210


      SUBROUTINE DRAWIT(X,Y,NDATA,RANGES,KKZ,MODCUR)                           33220
C*************************************************************************      33225
      DIMENSION GRID(61,81),XSCALE(5),YSCALE(7)                                33230
      DIMENSION X( 1), Y( 1), RANGES(4)                                        33240
      INTEGER*2 GRID,BLANK,DOT,XCHAR(4)/1H+,1H.,1H*,1HX/                       33250
      DATA DOT,BLANK/Z4B40,Z4040/                                              33260
C*************************************************************************      33265
      KDATA=NDATA*KKZ                                                          33270
      IF(MODCUR.GT.1) GO TO 444                                                33280
C*************************************************************************      33290
C     GRID IS THE MATRIX USED TO PLOT THE POINTS                         *     33300
C*************************************************************************      33310
      IERR=0                                                                   33320
      XMAX = RANGES(1)                                                         33330
      XMIN = RANGES(2)                                                         33340
      YMAX = RANGES(3)                                                         33350
      YMIN = RANGES(4)                                                         33360
C*************************************************************************      33370
C     CHECKING X AND Y POINTS AND PLOTTING THOSE CUT OF RANGE            *     33380
C     AT THE MARGIN                                                      *     33390
C*************************************************************************      33400
      DO 30 I=1,KDATA,KKZ                                                      33410
      IF(X(I).GT.XMAX.OR.X(I).LT.XMIN.OR.Y(I).GT.YMAX.OR.Y(I).LT.YMIN)         33420
```

```
     1  IERR=IERR+1                                                        33430
        IF(X(I).LE.XMAX) GO TO 205                                         33440
        X  (I)=XMAX                                                        33450
        GOTO 210                                                           33460
   205  IF(X(I).GE.XMIN) GO TO 210                                         33470
        X  (I)=XMIN                                                        33480
   210  IF(Y(I).LE.YMAX) GO TO 215                                         33490
        Y  (I)=YMAX                                                        33500
        GOTO 30                                                            33510
   215  IF(Y(I).GE.YMIN) GO TO 30                                          33520
        Y  (I)=YMIN                                                        33530
    30  CONTINUE                                                           33540
C*************************************************************************33550
C       PLOTTING X AND Y AXIS , IF NECESSARY                              *33560
C*************************************************************************33570
        JERR=0                                                             33580
        XRANGE=XMAX-XMIN                                                   33590
        YRANGE=YMAX-YMIN                                                   33600
        IF (YRANGE.NE.0.)  GO TO 298                                       33610
        IF(YMIN.EQ.0.) GO TO 889                                           33620
        YMIN=0.                                                            33630
        YRANGE=YMAX                                                        33640
        GO TO 299                                                          33650
   298  IF (XRANGE.NE.0.)  GO TO 299                                       33660
        IF(XMIN.EQ.0.) GO TO 887                                           33670
        XMIN=0.                                                            33680
        XRANGE=XMAX                                                        33690
C*************************************************************************33700
C       BLANKING OUT MATRIX-(GRID)                                        *33710
C*************************************************************************33720
   299  DO 300 I=1,61                                                      33730
        DO 301 JJ=1,81                                                     33740
301     GRID(I,JJ)=BLANK                                                   33750
   300  CONTINUE                                                           33760
        IF(XMAX*XMIN.GE.0.) GO TO 222                                      33770
        IYAXIS=80.*(-XMIN)/XRANGE+1.5                                      33780
        DO 40 I=1,61                                                       33790
    40  GRID(I,IYAXIS)=DOT                                                 33800
   222  IF(YMAX*YMIN.GE.0.) GO TO 333                                      33810
        IXAXIS=60.*YMAX/YRANGE+1.5                                         33820
        DO 60 I=1,81                                                       33830
    60  GRID(IXAXIS,I)=DOT                                                 33840
C*************************************************************************33850
C       COMPUTE PROPER SCALE NUMBERS                                      *33860
C*************************************************************************33870
   333  XINCR=XRANGE/4.                                                    33880
        YINCR=YRANGE/6.                                                    33890
        XSCALE(1)=XMAX                                                     33900
```

```
      XSCALE(5)=XMIN                                                    33910
      DO 80 I=2,4                                                       33920
      XSCALE(I)=XSCALE(I-1)-XINCR                                       33930
      IF(ABS(XSCALE(I)).LT.1.E-4) XSCALE(I)=0.                          33940
   80 CONTINUE                                                          33950
      YSCALE(1)=YMAX                                                    33960
      YSCALE(7)=YMIN                                                    33970
      DO 81 I=2,6                                                       33980
      YSCALE(I)=YSCALE(I-1)-YINCR                                       33990
      IF(ABS(YSCALE(I)).LT.1.E-4) YSCALE(I)=0.                          34000
   81 CONTINUE                                                          34010
      DO 85 II=1,2                                                      34020
      JJ=6-II                                                           34030
      XT=XSCALE(JJ)                                                     34040
      XSCALE(JJ)=XSCALE(II)                                             34050
   85 XSCALE(II)=XT                                                     34060
C*****************************************************************    34070
C*****************************************************************    34080
  444 IF(MODCUR.LT.2) JSET=0                                            34090
      IF(JERR.GT.0) GO TO 885                                           34100
      JSET=JSET+1                                                       34110
      IF(JSET.GT.4) JSET=1                                              34120
      DO 700 I=1,KDATA,KKZ                                              34130
      IPTX=60.*(YMAX-Y(I))/YRANGE+1.5                                   34140
      IPTY=80.*(X(I)-XMIN)/XRANGE+1.5                                   34150
      IF(IPTX.GT.61.OR.IPTY.GT.81) GO TO 70                             34160
      IF(IPTX.LE.0.OR.IPTY.LE.0)GO TO 70                                34170
      GRID(IPTX,IPTY) = XCHAR(JSET)                                     34180
      GO TO 700                                                         34190
   70 IERR=IERR+1                                                       34200
  700 CONTINUE                                                          34210
C*****************************************************************    34220
C     OUTPUT SECTION WITH GRAPH                                   *     34230
C*****************************************************************    34240
      IF(MODCUR.EQ.1.OR.MODCUR.EQ.2) RETURN                             34250
      AXR=ABS(XRANGE)                                                   34260
      AYR=ABS(YRANGE)                                                   34270
      IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 400                         34280
      WRITE(6,17) XSCALE                                                34290
   17 FORMAT(12X,1PE10.3,4(10X,E10.3)/15X,'**',8('+*********'),'+**')   34300
      GO TO 401                                                         34310
  400 WRITE(6,117) XSCALE                                               34320
  117 FORMAT( 8X,F11.2,4( 9X,F11.2)/15X,'**',8('+*********'),'+**')     34330
  401 II=1                                                              34340
      DO 101 IK=1,61                                                    34350
      IF(MOD(IK-1,10).NE.0) GO TO 92                                    34360
      IF(AYR.LT.1.E+8.AND.AYR.GE..95) GO TO 404                         34370
      WRITE(6,18) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)           34380
```

```
 18 FORMAT(3X,1PE10.3,2X,1H+,1X,81A1,1X,1H+,2X,E10.3)                    34390
    GO TO 405                                                            34400
404 WRITE(6,118) YSCALE(II),(GRID(IK,IX),IX=1,81),YSCALE(II)             34410
118 FORMAT(2X,F11.2,'   + ',81A1,' +   ',F11.2)                          34420
405 II=II+1                                                              34430
    GO TO 101                                                            34440
 92 WRITE(6,19) (GRID(IK,IX),IX=1,81)                                    34450
 19 FORMAT(15X,'* ',81A1,' *')                                           34460
101 CONTINUE                                                             34470
    IF(AXR.LT.1.E+8.AND.AXR.GE..95) GO TO 402                            34480
    WRITE(6,22) XSCALE                                                   34490
 22 FORMAT(15X,'**',8('+**********'),'+**'/12X,1PE10.3,4(10X,E10.3),//)   34500
    GO TO 403                                                            34510
402 WRITE(6,217) XSCALE                                                  34520
217 FORMAT(15X,'**',8('+**********'),'+**'/ 8X,F11.2,4( 9X,F11.2),//)     34530
403 IF(IERR.GT.0) WRITE(6,20) IERR                                       34540
 20  FORMAT(10X 'NUMBER OF POINTS OUT OF RANGE =',I4)                    34550
1000 RETURN                                                              34560
 889 WRITE(6,888)                                                        34570
 888 FORMAT(' ALL Y VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN Y  34580
    1 WHEN MODCUR=0 OR 1.')                                              34590
    JERR=10                                                              34600
    RETURN                                                               34610
 887 WRITE(6,886)                                                        34620
 886 FORMAT(' ALL X VALUES=0. CANNOT SETUP PLOT GRID. CHECK MAX & MIN X  34630
    1 WHEN MODCUR=0 OR 1.')                                              34640
    JERR=10                                                              34650
    RETURN                                                               34660
 885 WRITE(6,884)                                                        34670
 884 FORMAT(' GRID NOT SETUP WHEN MODCUR LAST 0 OR 1. NO PLOT UNTIL GRI  34680
    1D PROPERLY SETUP')                                                  34690
    RETURN                                                               34700
    END                                                                  34710
```

```
      SUBROUTINE RANGER (*)
C*****************************************************************
C     SUBROUTINE RANGER IS THE CONTROLLING SUBROUTINE FOR THE FREQUENCY*
C     RANGE SEARCH OPERATION. THE TOP AND BOTTOM OF THE RANGE IS CON-  *
C     VERTED FROM CPS TO -RADIANS/SEC AND A CHECK PERFORMED ON THE     *
C     LIMITS. THE LOWEST (SUB. BTTM) AND HIGHEST (SUB. TOPPER)         *
C     FREQUENCIES ARE DETERMINED PRIOR TO THE RANGE BEING PARTITIONED. *
C     THE ALGORITHM IS BIASED TOWARD THE LOWER FREQUENCIES. A FORM OF  *
C     THE AITKIN EXTRAPOLATION IS USED TO INCREASE THE CONVERGENCE     *
C     RATE. SUBROUTINE ADJUST IS CALLED FOR FINAL DISPOSITION OF EACH  *
C     PARTICULAR CASE.                                                 *
C*****************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,
     9SCALEW,SCALEX,SCALE3,VR,VI,WW,XX,YY,XAITKN,VALNEW,VALOLD,PCNT,
     8TWOPI,WV,XV,YV
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)
      COMMON /BL100/ TEEO,$DYNMC
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)
      COMMON /BLK2/ IVIBES,IVB
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP
      COMMON/BLK4/ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL
      COMMON/BLK5/$FLAG,$TAG,$DOVEC
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO
      COMMON/BLK7/SCALEX,SCALEW,SCALE3
      COMMON/BLK8/EPSVIB,EPSVEC,EPSAIT
      COMMON/BLK9/ITR1,ITR2
      COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS
      COMMON/BLK13/DIF1,DIF2,DIF3,DIF4,TOLLOW,TOLHI,TOLRNG
      XAITKN(WW,XX,YY) = WW - (((WW-XX)**2)/((WW + YY)-(2.0 * XX)))
C*****************************************************************
      BOTTOM = -((SHFTPT(1,IVB) * SNGL(TWOPI) * TEEO)**2)
      TOP    = -((SHFTPT(2,IVB) * SNGL(TWOPI) * TEEO)**2)
      CALL VECOUT(1,&1)
    1 IF(BOTTOM .LE. TOP) CALL VECOUT(4,&9)
      CALL BTTM(&9)
      CALL TOPPER(&9)
      IRGOLD = 1
    2 IF(IRGOLD.LE.0) CALL VECOUT(19,&9)
      I4 = 1
      VALNEW = ZERODO
      $TAG = .FALSE.
      IF(RANGE(I4,2).LE.(TOLRNG*RANGE(I4,3))) CALL ADJUST(1,&2,&2)
      $TAG = .TRUE.
      DELR = (RANGE(I4,3) - RANGE(I4,2))*TOLCLS
      ASHIFT = RANGE(I4,2) + DELR
```

```
      DELSHF = RANGE(I4,3) - DELR                                          35180
C*****************************************************************       35190
C     WRITE(6,333) I4,(RANGE(I4,J),J=1,4),(RANGE(I4+1,K),K=1,4)            35200
C 333 FORMAT(////35X,'RANGE',I2,' =',4E14.6//35X,'RANGE 2 =',4E14.6)       35210
C*****************************************************************       35220
    3 CALL START(2)                                                        35230
      CALL PMATRX                                                          35240
      DO 4 I5 = 1,ITR1                                                     35250
      $FLAG = .TRUE.                                                       35260
      CALL ITRATE                                                          35270
      VALOLD = VALNEW                                                      35280
      VALNEW = XAITKN(SCALEW,SCALEX,SCALE3)                                35290
       PCNT=DABS((DABS(VALNEW)-DABS(VALOLD))/VALNEW)                       35300
      IF(NJ.GT.ITR2) CALL VECOUT(7,&4)                                     35310
      IF(PCNT.LT. EPSAIT .AND.NJ.GT.3) GO TO 8                             35320
    4 CONTINUE                                                             35330
      CALL VECOUT(8,&5)                                                    35340
    5 CALL ITCPLX(&6)                                                      35350
      IF($FLAG) CALL ADJUST(5,&7,&2)                                       35360
    6 CALL ADJUST(1,&7,&2)                                                 35370
    7 $TAG = .FALSE.                                                       35380
      ASHIFT = DELSHF                                                      35390
      GO TO 3                                                              35400
    8 DIF1 = TOLLOW * RANGE(I4,1)                                          35410
      DIF2 = TOLHI * RANGE(I4,1)                                           35420
      VREAL = ONEDO / VALNEW                                               35430
      VREAL = DBLE(ASHFUM) + VREAL                                         35440
      AREAL = SNGL(VREAL)                                                  35450
      IF(AREAL.LE.DIF1.AND.AREAL.GE.DIF2) CALL ADJUST(2,&7,&2)             35460
      DIF3 = TOLLOW * RANGE(I4,4)                                          35470
      DIF4 = TOLHI * RANGE(I4,4)                                           35480
      IF(AREAL.LE.DIF3.AND.AREAL.GE.DIF4) CALL ADJUST(3,&7,&2)             35490
      IF(AREAL.LE.DIF2.AND.AREAL.GE.DIF3) CALL ADJUST(4,&7,&2)             35500
      IF(AREAL.GT.DIF1.OR.AREAL.LT.DIF4) CALL ADJUST(1,&7,&2)              35510
    9 RETURN 1                                                             35520
      END                                                                  35530


      SUBROUTINE ADJUST(IADJ,*,*)                                          35540
C*****************************************************************       35550
C     THIS SUBROUTINE, CALLED BY SUBROUTINE RANGER, PERFORMS THE FINAL *   35560
C     DISPOSITION OF EACH SEARCH OPERATION. WHEN A NEW REAL EIGENVALUE  *  35570
C     IS DETERMINED (IADJ=7 OR 12) IT IS REFINED, RECHECKED TO ENSURE   *  35580
C     THAT IT IS INDEED A NEW EIGENVALUE AND NEW RANGE PARAMETERS CALC-*   35590
C     ULATED FOR RANGES I AND I+1. IN THE EVENT THE EIGENVALUE FALLS    *  35600
C     OUT OF THE RANGE, HAS BEEN PREVIOUSLY DETERMINED OR IS COMPLEX    *  35610
C     THE TOTAL RANGE, OR A SUITABLE PORTION THEREOF, IS ELIMINATED.    *  35620
C*****************************************************************       35630
```

```
      IMPLICIT LOGICAL*1 ($)                                                    35640
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG                              35650
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)        35660
      COMMON/BLK4/ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL                   35670
      COMMON/BLK5/$FLAG,$TAG,$DOVEC                                             35680
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO                                 35690
      COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS                  35700
      COMMON/BLK13/DIF1,DIF2,DIF3,DIF4,TOLLOW,TOLHI,TOLRNG                      35710
C*****************************************************************************35720
      GO TO (1,3,4,7,12),IADJ                                                   35730
    1 IF($TAG) RETURN 1                                                         35740
    5 DO 2 J = 1,4                                                              35750
      DO 2 I = I4,IRGOLD                                                        35760
      IP1 = I+ 1                                                                35770
    2 RANGE(I,J) = RANGE(IP1,J)                                                 35780
      IRGOLD = IRGOLD - 1                                                       35790
      RETURN 2                                                                  35800
    3 RANGE(I4,2) = ASHFUM + (ASHFUM - RANGE(I4,1))                             35810
      IF($TAG) RETURN 1                                                         35820
      GO TO 6                                                                   35830
    4 RANGE(I4,3) = ASHFUM - (RANGE(I4,4) - ASHFUM)                             35840
    6 IF(RANGE(I4,2).LE.(TOLRNG*RANGE(I4,3))) GO TO 5                           35850
      IF($TAG) RETURN 1                                                         35860
      RETURN 2                                                                  35870
    7 ASHIFT = ASHFUM + TOLSHF * (AREAL - ASHFUM)                               35880
      ARNG1 = ASHFUM                                                            35890
      CALL PMATRX                                                               35900
      CALL ITREAL(&8)                                                           35910
      CALL ITCPLX(&1)                                                           35920
      GO TO 1                                                                   35930
    8 ASHFUM = ARNG1                                                            35940
       DIF1 = TOLLOW * RANGE(I4,1)                                              35950
       DIF2 = TOLHI * RANGE(I4,1)                                               35960
       DIF3 = TOLLOW * RANGE(I4,4)                                              35970
       DIF4 = TOLHI * RANGE(I4,4)                                               35980
      IF(AREAL.LE.DIF1.AND.AREAL.GE.DIF2) GO TO 3                               35990
      IF(AREAL.LE.DIF3.AND.AREAL.GE.DIF4) GO TO 4                               36000
      IF(AREAL.GT.DIF1.OR.AREAL.LT.DIF4) GO TO 1                                36010
      CALL VECOUT(9,&9)                                                         36020
    9 DO 10 I = 1,4                                                             36030
      IM1 = IRGOLD + 2                                                          36040
      DO 10 J = I4,IRGOLD                                                       36050
      IM1 = IM1 - 1                                                             36060
      IM2 = IM1 - 1                                                             36070
   10  RANGE(IM1,I) = RANGE(IM2,I)                                              36080
      IRGOLD = IRGOLD + 1                                                       36090
      NUMBER = NUMBER + 1                                                       36100
      RANGE(I4,3) = AREAL                                                       36110
```

```
      RANGE(I4,4) = AREAL                                                    36120
      RANGE(I4+1,1) = AREAL                                                  36130
      RANGE(I4+1,2) = AREAL                                                  36140
      IF(RANGE(I4,2).LE.(TOLRNG*RANGE(I4,3))) GO TO 5                        36150
      GO TO (11,14,16),ITWO                                                  36160
   11 RETURN 2                                                               36170
   12 DIF1 = TOLLOW * RANGE(I4,1)                                            36180
      DIF2 = TOLHI * RANGE(I4,1)                                             36190
      DIF3 = TOLLOW * RANGE(I4,4)                                            36200
      DIF4 = TOLHI * RANGE(I4,4)                                             36210
      AREAL = SNGL(VREAL)                                                    36220
      BREAL = SNGL(VIMAG)                                                    36230
      IF((BREAL.GE.DIF4.AND.BREAL.LE.DIF3).AND.(AREAL.GE.DIF2.AND.AREAL.     36240
     1LE.DIF1)) GO TO 5                                                      36250
      IF(BREAL.GE.DIF4.AND.BREAL.LE.DIF3) RANGE(I4,3) = ASHFUM               36260
      IF(AREAL.GE.DIF2.AND.AREAL.LE.DIF1) RANGE(I4,2) = ASHFUM               36270
      IF((AREAL.GT.DIF1.OR.AREAL.LT.DIF4).AND.(BREAL.LT.DIF4.OR.BREAL.GT     36280
     1.DIF1)) GO TO 1                                                        36290
      IF((AREAL.LT.DIF2.AND.AREAL.GT.DIF3).AND.(BREAL.LT.DIF2.AND.BREAL      36300
     1.GT.DIF3)) GO TO 13                                                    36310
      IF((AREAL.LT.DIF4.OR.AREAL.GT.DIF1).AND.(BREAL.GT.DIF3.AND.BREAL.L     36320
     1T.DIF2)) GO TO 18                                                      36330
      IF((BREAL.LT.DIF4.OR.BREAL.GT.DIF1).AND.(AREAL.GT.DIF3.AND.AREAL.L     36340
     1T.DIF2)) GO TO 17                                                      36350
      RETURN 2                                                               36360
   13 DIST1 = ABS(ASHFUM - BREAL)                                            36370
      DIST2 = ABS(AREAL - ASHFUM)                                            36380
      ATEMP = AREAL                                                          36390
      IF(DIST2.GT.DIST1) GO TO 15                                            36400
      ITWO = 2                                                               36410
      GO TO 7                                                                36420
   14 AREAL = BREAL                                                          36430
      I4 = I4 + 1                                                            36440
      CALL START(2)                                                          36450
      GO TO 7                                                                36460
   15 AREAL = BREAL                                                          36470
      ITWO = 3                                                               36480
      GO TO 7                                                                36490
   16 AREAL = ATEMP                                                          36500
      CALL START(2)                                                          36510
      GO TO 7                                                                36520
   17 ITWO = 1                                                               36530
      GO TO 7                                                                36540
   18 AREAL = BREAL                                                          36550
      ITWO = 1                                                               36560
      GO TO 7                                                                36570
      END                                                                    36580
```

```
      SUBROUTINE BTTM(*)                                                36590
C*****************************************************************     36600
C     SUBROUTINE BTTM DETERMINES THE LOWEST FREQUENCY IN THE OVERALL   *  36610
C     RANGE. THE CONVERGENCE TEST ON THE VECTOR IS ALWAYS PERFORMED.   *  36620
C     A FORM OF THE AITKIN EXTRAPOLATION IS USED TO INCREASE THE CON-  *  36630
C     VERGENCE RATE. THE RANGE WILL BE SEARCHED BOTTOM TO TOP IN PRE-  *  36640
C     DETERMINED SEGMENTS UNTIL  A REAL EIGENVALUE IS DETERMINED AS    *  36650
C     THE BOTTOM ( TOLUP AND ATRY IN SUBROUTINE INITAA ).              *  36660
C*****************************************************************     36670
      IMPLICIT LOGICAL*1 ($)                                            36680
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,36690
     9SCALEW,SCALEX,SCALE3,VR,VI,WW,XX,YY,XAITKN,TEMP1,TEMP2,TEMP3,     36700
     8TWOPI,CLDVR                                                       36710
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)36720
      COMMON/BLK4/ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL           36730
      COMMON/BLK5/$FLAG,$TAG,$DOVEC                                     36740
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO                         36750
      COMMON/BLK7/SCALEX,SCALEW,SCALE3                                  36760
      COMMON/BLK8/EPSVIB,EPSVEC,EPSAIT                                  36770
      COMMON/BLK9/ITR1,ITR2                                             36780
      COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS          36790
      COMMON /BLK12/NTRY                                                36800
      SIZE(VR,VI) = - SNGL(DSQRT(VR*VR + VI*VI))                        36810
      XAITKN(WW,XX,YY) = WW - (((WW-XX)**2)/((WW + YY)-(2.0 * XX)))     36820
C*****************************************************************     36830
      $TAG = $DOVEC                                                     36840
      $DOVEC = .TRUE.                                                   36850
      NI = 1                                                            36860
      DELSHF = TOLUP  * (TOP - BOTTOM)                                  36870
      DO 8 I = 1,4                                                      36880
    8 RANGE(2,I) = 0.0                                                  36890
    1 IF(BOTTOM.LE.TOP) CALL VECOUT(11,&5)                              36900
      IF(NI.GT.NTRY) CALL VECOUT(12,&5)                                 36910
      NI = NI + 1                                                       36920
      CALL START(1)                                                     36930
      ASHIFT = BOTTOM                                                   36940
      CALL PMATRX                                                       36950
      DO 9 I = 1,ITR1                                                   36960
      $FLAG = .TRUE.                                                    36970
      CALL ITRATE                                                       36980
      TEMP1 = XAITKN(SCALEW,SCALEX,SCALE3)                              36990
      TEMP2 = CABS(TEMP1)                                               37000
      TEMP3 = CABS((TEMP2 - CLDVR)/TEMP2)                               37010
      CLDVR = TEMP2                                                     37020
      IF(NJ.GT.ITR2) CALL VECOUT(7,&9)                                  37030
      IF(TEMP3.GE.EPSAIT)   GO TO 9                                     37040
      IF(NJ.LT.3) GO TO 9                                               37050
      VREAL = (ONEDO/TEMP1) + DBLE(ASHFUM)                              37060
```

```
      ASHIFT = ASHFUM + TOLSHF * (SNGL(VREAL) - ASHFUM)                 37070
      IF(ABS(ASHIFT).LE.1.0) GO TO 11                                   37080
      CALL PMATRX                                                       37090
   11 CALL ITREAL(&6)                                                   37100
      GO TO 10                                                          37110
    9 CONTINUE                                                          37120
      CALL VECOUT(8,&10)                                                37130
   10 ARTSIZ = 0.0                                                      37140
      CALL ITCPLX(&2)                                                   37150
      IF($FLAG) GO TO 6                                                 37160
      ARTSIZ = SIZE(VREAL,VIMAG)                                        37170
    2 BOTTOM = BOTTOM + DELSHF                                          37180
      IF(ARTSIZ .LT. BOTTOM) BOTTOM = BOTTOM - DELSHF + 1.1 * ARTSIZ    37190
      CALL VECOUT(10,&1)                                                37200
    6 AREAL = SNGL(VREAL)                                               37210
      AIMAG = SNGL(VIMAG)                                               37220
      DIST1 = ABS(AIMAG - BOTTOM)                                       37230
      DIST2 = ABS(AREAL - BOTTOM)                                       37240
      IF(DIST1.LT.DIST2.AND.VIMAG.LT.ZERODO) AREAL = AIMAG              37250
      RANGE(1,1) = AREAL                                                37260
      NUMBER = 1                                                        37270
      $DOVEC = $TAG                                                     37280
      IF( AREAL .LE. BOTTOM) RANGE(1,2) = AREAL                         37290
      IF( AREAL .GT. BOTTOM) RANGE(1,2) = BOTTOM                        37300
      CALL VECOUT(20,&3)                                                37310
    3 IF(VIMAG .EQ. ZERODO) CALL VECOUT(9,&7)                           37320
    7 IF(RANGE(1,2) .LE. TOP) CALL VECOUT(11,&5)                        37330
    4 RETURN                                                            37340
    5 RETURN 1                                                          37350
      END                                                               37360

      SUBROUTINE TOPPER(*)                                              37370
C*****************************************************************     37380
C     SUBROUTINE TOPPER DETERMINES THE HIGHEST FREQUENCY IN THE OVERALL*  37390
C     RANGE. IT OPERATES IN A MANNER SIMILAR TO SUBROUTINE BTTM.    *   37400
C*****************************************************************     37410
      IMPLICIT LOGICAL*1 ($)                                            37420
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,37430
     9SCALEW,SCALEX,SCALE3,VR,VI,UW,XX,YY,XAITKN,TEMP1,TEMP2,TEMP3,     37440
     8TWOPI,OLDVR                                                       37450
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)37460
      COMMON/BLK4/ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL           37470
      COMMON/BLK5/$FLAG,$TAG,$DOVEC                                     37480
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO                         37490
      COMMON/BLK7/SCALEX,SCALE2,SCALE3                                  37500
      COMMON/BLK8/EPSVIB,EPSVEC,EPSAIT                                  37510
      COMMON/BLK9/ITR1,ITR2                                             37520
```

```
      COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS                37530
      COMMON /BLK12/NTRY                                                      37540
      SIZE(VR,VI) = - SNGL(DSQRT(VR*VR + VI*VI))                              37550
      XAITKN(WW,XX,YY) = WW - (((WW-XX)**2)/((WW + YY)-(2.0 * XX)))           37560
C*****************************************************************          37570
      IF(TOP .EQ. 0.0) TOP =-1000.0                                           37580
      NI = 1                                                                  37590
    1 IF(RANGE(1,2).LE.TOP) CALL VECOUT(13,&5)                                37600
      IF(NI.GT.NTRY) CALL VECOUT(14,&5)                                       37610
      NI = NI + 1                                                             37620
      CALL START(2)                                                           37630
      ASHIFT = TOP                                                            37640
      CALL PMATRX                                                             37650
      DO 9 I = 1,ITR1                                                         37660
      $FLAG = .TRUE.                                                          37670
      CALL ITRATE                                                             37680
      TEMP1 = XAITKN(SCALEW,SCALEX,SCALE3)                                    37690
      TEMP2 = DABS(TEMP1)                                                     37700
      TEMP3 = DABS((TEMP2 - OLDVR)/TEMP2)                                     37710
      OLDVR = TEMP2                                                           37720
      IF(NJ.GT.ITR2) CALL VECOUT(7,&9)                                        37730
      IF(TEMP3.GE.EPSAIT)   GO TO 9                                           37740
      IF(NJ.LT.3) GO TO 9                                                     37750
      VREAL = (ONEDO/TEMP1) + DBLE(ASHFUM)                                    37760
      ASHIFT = ASHFUM + TOLSHF * (SNGL(VREAL) - ASHFUM)                       37770
      CALL PMATRX                                                             37780
      CALL ITREAL(&6)                                                         37790
      GO TO 10                                                                37800
    9 CONTINUE                                                                37810
      CALL VECOUT(8,&10)                                                      37820
   10 ARTSIZ = 0.0                                                            37830
      CALL ITCPLX(&2)                                                         37840
      IF($FLAG) GO TO 6                                                       37850
      ARTSIZ = SIZE(VREAL,VIMAG)                                              37860
    2 TOP = TOP - DELSHF                                                      37870
      IF(ARTSIZ .LT. TOP) TOP = TOP + DELSHF - 1.1 * ARTSIZ                   37880
      CALL VECOUT(15,&1)                                                      37890
    6 AREAL = SNGL(VREAL)                                                     37900
      AIMAG = SNGL(VIMAG)                                                     37910
      DIST1 = ABS(AIMAG - TOP)                                                37920
      DIST2 = ABS(AREAL - TOP)                                                37930
      IF(DIST1.LT.DIST2.AND.VIMAG.LT.ZERODO) AREAL = AIMAG                    37940
      RANGE(1,4) = AREAL                                                      37950
      NUMBER = 2                                                              37960
      IF( AREAL .GT. TOP) RANGE(1,3) = AREAL                                  37970
      IF(AREAL .LE. TOP) RANGE(1,3) = TOP                                     37980
      CALL VECOUT(21,&3)                                                      37990
    3 IF(VIMAG .EQ. ZERODO) CALL VECOUT(9,&7)                                 38000
```

```
      7 IF(RANGE(1,2).LE.RANGE(1,3)) CALL VECOUT(11,&5)                  38010
      4 RETURN                                                          38020
      5 RETURN 1                                                        38030
        END                                                             38040

        SUBROUTINE ITREAL(*)                                            38050
C*******************************************************************  38060
C     THIS SUBROUTINE DIRECTS THE ITERATION PROCESS THAT DETERMINES    *38070
C     REAL EIGENVALUES AND VECTORS. THE CONVERGENCE TEST ON THE EIGEN- *38080
C     VALUES IS PERFORMED IN THIS SUBROUTINE.                          *38090
C*******************************************************************  38100
        IMPLICIT LOGICAL*1 ($)                                          38110
        REAL*8 ZEROOO,PTFIVE,ONEOO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,38120
       1SCALEW,SCALEX,SCALE3,DELVR,OLDVR,TEMP,WV,XV,YV                  38130
        COMMON /IBL12/ KMAX1,KMAX2,NCONV                                38140
        COMMON /BLK1/ VMASS,NN2,$VIBES                                  38150
        COMMON/BLK4/ZEROOO,PTFIVE,ONEOO,SMALL,VREAL,VIMAG,AREAL         38160
        COMMON/BLK5/$FLAG,$TAG,$DOVEC                                   38170
        COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO                       38180
        COMMON/BLK7/SCALEX,SCALEW,SCALE3                                38190
        COMMON/BLK8/EPSVIB,EPSVEC,EPSAIT                                38200
        COMMON/BLK9/ITR1,ITR2                                           38210
        COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,DII,TOLUP,TOLDWN,TOLCLS        38220
C*******************************************************************  38230
        $FLAG = .TRUE.                                                  38240
        NN2 = 2                                                         38250
        DO 17 I1 = 1,ITR1                                               38260
        CALL ITRATE                                                     38270
        IF(NJ .GT. ITR2) GO TO 20                                       38280
        TEMP = DABS(SCALEW)                                             38290
        DELVR = DABS((TEMP - OLDVR)/TEMP)                               38300
        OLDVR = TEMP                                                    38310
        IF(DELVR.GT.EPSVIB) GO TO 17                                    38320
        IF($DOVEC .AND. (NCONV .EQ. 0)) GO TO 17                        38330
        NN2 = 1                                                         38340
        CALL ITRATE                                                     38350
        VREAL = ONEOO / SCALEW                                          38360
        VREAL = VREAL + DBLE(ATSHF)                                     38370
        AREAL = SNGL(VREAL)                                             38380
        RETURN 1                                                        38390
     20 CONTINUE                                                        38400
        VREAL = ONEOO / SCALEW                                          38410
        VREAL = VREAL + DBLE(ATSHF)                                     38420
        AREAL = SNGL(VREAL)                                             38430
        CALL VECOUT(7,&17)                                              38440
     17 CONTINUE                                                        38450
        VREAL = ONEOO / SCALEW                                          38460
```

```
      VREAL = VREAL + DBLE(ATSHF)                                       38470
      AREAL = SNGL(VREAL)                                               38480
      CALL VECOUT(8,&19)                                                38490
   19 RETURN                                                            38500
      END                                                               38510

      SUBROUTINE ITRATE                                                 38520
C*****************************************************************     38530
C     THIS SUBROUTINE PERFORMS THE ITERATION COMPUTATIONS.  POTTERS'  *  38540
C     SCHEME IS USED (EQUATIONS 29A AND 29B OF REF. 1). THE RAYLEIGH  *  38550
C     QUOTIENT EIGENVALUE IS CALCULATED AND THE VECTOR SCALED. IN     *  38560
C     ADDITION IF THE VECTOR CONVERGENCE TEST IS TO BE PERFORMED IT IS *  38570
C     PERFORMED IN THIS SUBROUTINE.                                   *  38580
C*****************************************************************     38590
      IMPLICIT LOGICAL*1 ($)                                            38600
      REAL*8 EPSVIB,EPSVEC,EPSAIT,ONEDO,PTFIVE,SCALEW,SCALEX,SCALE3,    38610
     1       SMALL,SUM,SUM1,TEMPW,TEMPX,TWOPI,VREAL,VIMAG,WV,XV,YV,     38620
     2       ZERODO,ZT,ZV,TEMPV,TEMPY,WT,XT,YT                          38630
      COMMON /IBL3/ MO,M1,M2,M3                                         38640
      COMMON /IBL4/ KMAX,KL                                             38650
      COMMON /IBL5/ IBCINL,IBCFNL                                       38660
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                  38670
      COMMON /BL1/ A(4,4),BEE(4,4),C(4,4)                               38680
      COMMON /BL4/ P(4,4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3M(4,4,10),   38690
     1             ZF4M(4,4,10)                                         38700
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)38710
      COMMON /BL34/ DEE(4,4,200),DST(4,4,200)                           38720
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)38730
      COMMON /BLK1/ VMASS,NN2,$VIBES                                    38740
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP                         38750
      COMMON /BLK4/ ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL         38760
      COMMON /BLK5/ $FLAG,$TAG,$DOVEC                                   38770
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO                         38780
      COMMON/BLK7/SCALEX,SCALEW,SCALE3                                  38790
      COMMON /BLK8/ EPSVIB,EPSVEC,EPSAIT                                38800
      COMMON /BLK11/ DIAG(4)                                            38810
      DIMENSION ZV(4,200),ZT(4),ST(4),IPIVOT(4),INDEX(4,2),CLO(4,4),    38820
     4          CL1(4,4),CL2(4,4),WT(812),XT(812),YT(812)               38830
      EQUIVALENCE (FILL(1),ZT(1)),(FILL(9),ST(1)),(CLO(1,1),ZF1M(1,1,1))38840
     3           ,(CL1(1,1),ZF2M(1,1,1)),(CL2(1,1),ZF3M(1,1,1)),       38850
     2           (WV(1,1),WT(1)),(XV(1,1),XT(1)),(YV(1,1),YT(1))        38860
C*****************************************************************     38870
C     INCREMENT THE ITERATION COUNTERS.                               *  38880
C*****************************************************************     38890
      NJ = NJ + 1                                                       38900
      NK = NK + 1                                                       38910
C*****************************************************************     38920
```

```
C     RETAIN THE LAST TWO SOLUTIONS.                                    *
C*********************************************************************
      SCALE3 = SCALEX
      SCALEX = SCALEW
      DO 102 I = IGO,ISTOP
  102 YT(I) = XT(I)
      DO 103 I = IGO,ISTOP
  103 XT(I) = WT(I)
C*********************************************************************
C     MULTIPLY THE RIGHT HAND SIDE VECTOR BY THE DIAGONAL              *
C     OF THE MASS MATRIX (WV(J,203),J=1,4).                            *
C*********************************************************************
      DO 104 I = IBEGIN,IEND
      DO 104 J = 1,4
  104 WV(J,I) = WV(J,I) * WV(J,203)
C*********************************************************************
C     BEGIN FORWARD PASS.                                              *
C      INITIAL POLE BOUNDARY CONDITIONS ON ZV ARE ZV(J,1)=0 FOR J=1,4. *
C*********************************************************************
      IF(IBCINL .GE. 0) GO TO 106
C*********************************************************************
      DO 105 J = 1,4
  105 ZV(J,1) = ZERODO
      GO TO 4
C*********************************************************************
  106 DO 3 I = 1,4
      SUM = ZERODO
      ZT(I) = WV(I,2)
      DO 2 J = 1,4
    2 SUM = SUM + DBLE(DG(I,J,1)) * WV(J,2)
    3 ZV(I,1) = SUM
C*********************************************************************
C     FORWARD PASS LOOP.                                               *
C*********************************************************************
    4 DO 7 K = 2,KL
      KM1 = K - 1
      KP1 = K + 1
      DO 6 I = 1,4
      SUM = ZERODO
      DO 5 J = 1,4
    5 SUM= SUM + DBLE(DEE(I,J,K))*WV(J,KP1) - DBLE(DST(I,J,K))*ZV(J,KM1)
    6 ZV(I,K) = SUM
    7 CONTINUE
C*********************************************************************
C     COMPLETE FORWARD PASS AND BEGIN BACKWARD PASS.                   *
C     FINAL POLE BOUNDARY CONDITIONS                                   *
C     ARE HANDLED IN STATEMENTS 20 THRU 29.                            *
C*********************************************************************
```

```
      IF(IBCFNL .LT. 0) GO TO 20                                        39410
C*****************************************************************     39420
      KM = KMAX                                                         39430
      DO 9 I = 1,4                                                      39440
      SUM = ZERODO                                                      39450
      DO 8 J = 1,4                                                      39460
    8 SUM=SUM+DBLE(DEE(I,J,KM))*WV(J,KMAX1)-DBLE(DST(I,J,KM))*ZV(J,KL)  39470
    9 ZV(I,KMAX) = SUM                                                  39480
C*****************************************************************     39490
      DO 11 I = 1,4                                                     39500
      SUM = ZERODO                                                      39510
      DO 10 J = 1,4                                                     39520
      SUM = SUM + DBLE(ZF2M(I,J,1)) * ZV(J,KMAX)                        39530
   10 SUM = SUM + DBLE(ZF3M(I,J,1)) * ZV(J,KL)                          39540
   11 WV(I,KMAX2) = SUM                                                 39550
      ISTART = 1                                                        39560
C*****************************************************************     39570
C     BACKWARD PASS LOOP.                                               39580
C*****************************************************************     39590
   12 DO 15 KK = ISTART,KMAX                                            39600
      K = KMAX1 - KK                                                    39610
      KP1 = K + 1                                                       39620
      KP2 = K + 2                                                       39630
      DO 14 I = 1,4                                                     39640
      SUM = ZERODO                                                      39650
      DO 13 J = 1,4                                                     39660
   13 SUM = SUM - DBLE(P(I,J,K)) * WV(J,KP2)                            39670
      SUM = SUM + ZV(I,K)                                               39680
   14 WV(I,KP1) = SUM                                                   39690
   15 CONTINUE                                                          39700
C*****************************************************************     39710
C     COMPLETE BACKWARD PASS ( COMPLETES SOLUTION VECTOR WV ).          39720
C*****************************************************************     39730
      IF(IBCINL .LT. 0) GO TO 30                                        39740
C*****************************************************************     39750
      CALL EFG(1,1)                                                     39760
      CALL ABC                                                          39770
      DO 18 I = 1,4                                                     39780
      SUM = ZERODO                                                      39790
      DO 17 J = 1,4                                                     39800
   17 SUM = SUM - DBLE(A(I,J))* WV(J,3) - DBLE(BEE(I,J))* WV(J,2)       39810
   18 ST(I) = SNGL(SUM + ZT(I))                                         39820
      CALL MATINV(C,4,ST,1,DETERM,IPIVOT,INDEX,4,ISCALE)                39830
      DO 19 I = 1,4                                                     39840
      WV(I,1) = DBLE(ST(I))                                             39850
   19 ZT(I) = ZERODO                                                    39860
      GO TO 30                                                          39870
C*****************************************************************     39880
```

```
C       THE FOLLOWING BLOCK OF STATEMENTS  DETERMINE WV(J,KMAX1)          *    39890
C       FOR A SHELL WITH A FINAL POLE.                                    *    39900
C*****************************************************************************39910
   20 DO 201 J = 1,4                                                           39920
  201 WV(J,KMAX1) = ZERODO                                                     39930
      IF(M2.EQ.0) GO TO 23                                                     39940
      DO 22 I = 1,4                                                            39950
      SUM = ZERODO                                                             39960
      DO 21 J = 1,4                                                            39970
   21 SUM = SUM + DBLE(CL2(I,J)) * ZV(J,KL)                                    39980
   22 WV(I,KMAX1) = SUM                                                        39990
   23 IF(M1.EQ.0) GO TO 26                                                     40000
      DO 25 I = 1,4                                                            40010
      SUM = ZERODO                                                             40020
      DO 24 J = 1,4                                                            40030
   24 SUM = SUM + DBLE(CL1(I,J)) * ZV(J,KL)                                    40040
   25 WV(I,KMAX1) = SUM                                                        40050
   26 IF(M0.EQ.0) GO TO 29                                                     40060
      DO  28 I = 1,4                                                           40070
      SUM = ZERODO                                                             40080
      DO 27 J = 1,4                                                            40090
   27 SUM = SUM + DBLE(CL0(I,J)) * ZV(J,KL)                                    40100
   28 WV(I,KMAX1) = SUM                                                        40110
   29 ISTART = 2                                                               40120
      GO TO 12                                                                 40130
C*****************************************************************************40140
C*****************************************************************************40150
C       CALCULATE EIGENVALUE AND SCALE SOLUTION VECTOR.                   *    40160
C*****************************************************************************40170
   30 TEMPV = ZERODO                                                           40180
      IF($FLAG) GO TO 302                                                      40190
C*****************************************************************************40200
C       IF $FLAG IS FALSE ITRATE CALLED IN ITCPLX.                        *    40210
C       DETERMINE AND SCALE BY THE MAXIMUM(ABSOLUTE) ELEMENT AND RETURN.  *    40220
C*****************************************************************************40230
      DO 301 I = IGO,ISTOP                                                     40240
      TEMPW = WT(I)                                                            40250
      TEMPX = DABS(TEMPW)                                                      40260
      IF(TEMPX .LE. TEMPV) GO TO 301                                           40270
      TEMPV = TEMPX                                                            40280
      TEMPY = TEMPW                                                            40290
  301 CONTINUE                                                                 40300
      SCALEW = TEMPY                                                           40310
      TEMPW = ONEDO / TEMPY                                                    40320
      GO TO 31                                                                 40330
C*****************************************************************************40340
C       IF $FLAG IS TRUE ITRATE CALLED IN ITREAL,RANGER,BTTM OR TOPPER.   *    40350
C       CALCULATE RAYLEIGH QUOTIENT EIGENVALUE AND                        *    40360
```

```
C      DETERMINE THE MAXIMUM(ABSOLUTE) ELEMENT.                          *   40370
C*************************************************************************   40380
  302 SUM = ZERODO                                                           40390
      SUM1= ZERODO                                                           40400
      DO 303 I = IBEGIN,IEND                                                 40410
      DO 303 J = 1,4                                                         40420
      TEMPW = WV(J,I)                                                        40430
      TEMPX = XV(J,I) * WV(J,203)                                            40440
      SUM = SUM + TEMPW * TEMPX                                              40450
      SUM1 = SUM1 + TEMPX * TEMPX                                            40460
      TEMPX = CABS(TEMPW)                                                    40470
      IF(TEMPX  .LE. TEMPV) GO TO 303                                        40480
      TEMPV = TEMPX                                                          40490
      TEMPY = TEMPW                                                          40500
  303 CONTINUE                                                               40510
      TEMPW = ONEDO / TEMPY                                                  40520
      SCALEW = SUM / SUM1                                                    40530
C*************************************************************************   40540
C      IF NN2 = 2 SCALE BY MAXIMUM ELEMENT AND RETURN.                   *   40550
C*************************************************************************   40560
      GO TO (304,31),NN2                                                     40570
C*************************************************************************   40580
C      IF NN2 = 1 SOLUTION HAS CONVERGED TO A REAL EIGENVALUE.           *   40590
C      SCALE VECTOR BY RAYLEIGH QUOTIENT EIGENVALUE.                     *   40600
C      COMPUTE DIFFERENCE BETWEEN THIS AND NEXT TO LAST SOLUTION ( THE   *   40610
C      STARTING VECTOR FOR THE NEXT EIGENVALUE  ). RETURN.               *   40620
C*************************************************************************   40630
  304 TEMPW = SUM1 / SUM                                                     40640
      NN2 = 2                                                                40650
      VIMAG = ZERODO                                                         40660
      DO 305 I = IGO,ISTOP                                                   40670
      WT(I) = WT(I) * TEMPW                                                  40680
  305 YT(I) = YT(I) - WT(I)                                                  40690
      RETURN                                                                 40700
C*************************************************************************   40710
   31 NCONV = 1                                                              40720
      VIMAG = ZERODO                                                         40730
C*************************************************************************   40740
C      SCALE VECTOR BY THE MAXIMUM ELEMENT.                              *   40750
C*************************************************************************   40760
      DO 311 I = IGO,ISTOP                                                   40770
  311 WT(I) = WT(I) * TEMPW                                                  40780
C*************************************************************************   40790
C       THE FOLLOWING BLOCK OF STATEMENTS PERFORM THE CONVERGENCE        *   40800
C       TEST ON THE EIGENVECTOR. EXECUTED WHEN $DOVEC IS TRUE.           *   40810
C*************************************************************************   40820
      IF(.NOT.$DOVEC) RETURN                                                 40830
C*************************************************************************   40840
```

```
C     CALCULATE  CONVERGENCE CRITERION.                                  *
C*********************************************************************
      DO 32 I = IBEGIN,IEND
      DO 32 J = 1,4
      ZT(J) = DMAX1(DABS(WV(J,I)),ZT(J))
   32 CONTINUE
      DO 33 I = 1,4
   33 ZT(I) = EPSVEC * ZT(I)
C*********************************************************************
C     PERFORM THE TEST.                                                 *
C*********************************************************************
      DO 34 I = IBEGIN,IEND
      DO 34 J = 1,4
      TEMPW = DABS(WV(J,I))
      IF(NCONV.EQ.0 .OR. TEMPW.LT.(1.D-03)) GO TO 34
      TEMPX = DABS(TEMPW - DABS(XV(J,I)))
      IF(TEMPX .GE. ZT(J))  GO TO 35
   34 CONTINUE
C*********************************************************************
C     PASSED VECTOR CONVERGENCE TEST.                                   *
C*********************************************************************
      RETURN
C*********************************************************************
C     FAILED VECTOR CONVERGENCE TEST.                                   *
C*********************************************************************
   35 NCONV = 0
      RETURN
      END


      SUBROUTINE ITCPLX(*)
C*********************************************************************
C     IN THE EVENT OF NO CONVERGENCE TO A REAL EIGENVALUE IN THE        *
C     SPECIFIED NUMBER OF ITERATIONS CONTROL IS PASSED TO THIS SUB-     *
C     ROUTINE. THE ALGORITHM ATTEMPTS TO FIND THE TWO EIGENVALUES       *
C     CLOSEST TO THE SHIFT POINT. THE METHOD OF LEAST SQUARES IS USED   *
C     TO DETERMINE THE COEFFICIENTS OF THE QUADRATIC EQUATION WHOSE     *
C     ASSUMED ROOTS ARE THE EIGENVALUES SOUGHT. THEY MAY BE REAL OR A   *
C     COMPLEX CONJUGATE PAIR.                                           *
C*********************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,
     1SCALEW,SCALEX,SCALE3,OLDVR,OLDVI,RAD,TEMP,DELVR,DELVI,ZA,ZB,ZC,
     2B,C,D,E,F,G,H,P,Q,U,WV,XV,YV,WT,XT,YT,TWOPI
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP
      COMMON/BLK4/ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL
      COMMON/BLK5/$FLAG,$TAG,$DOVEC
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO
```

```
      COMMON/BLK7/SCALEX,SCALEW,SCALE3                                          41310
      COMMON/BLK8/EPSVIB,EPSVEC,EPSAIT                                          41320
      COMMON/BLK9/ITR1,ITR2                                                     41330
      COMMON/BLK10/ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS                  41340
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)        41350
      DIMENSION WT(812),XT(812),YT(812)                                         41360
      EQUIVALENCE (WV(1,1),WT(1)),(XV(1,1),XT(1)),(YV(1,1),YT(1))               41370
C*****                                                                          41380
      CALL START(1)                                                             41390
      OLDVR = ZERODO                                                            41400
      OLDVI = ZERODO                                                            41410
      SCALEW = ONEDO                                                            41420
      DO 5 I1 = 1,ITR1                                                          41430
      $FLAG = .FALSE.                                                           41440
      CALL ITRATE                                                               41450
      B = ZERODO                                                                41460
      C = ZERODO                                                                41470
      D = ZERODO                                                                41480
      F = ZERODO                                                                41490
      G = ZERODO                                                                41500
      DO 1 I = IGO,ISTOP                                                        41510
    1 B = B + XT(I) * XT(I)                                                     41520
      DO 2 I = IGO,ISTOP                                                        41530
    2 C = C + XT(I) * YT(I)                                                     41540
      DO 3 I = IGO,ISTOP                                                        41550
    3 D = D + XT(I) * WT(I)                                                     41560
      DO 4 I = IGO,ISTOP                                                        41570
    4 F = F + YT(I) * YT(I)                                                     41580
      DO 9 I = IGO,ISTOP                                                        41590
    9 G = G + YT(I) * WT(I)                                                     41600
      B = B * SCALEX                                                            41610
      E = C * SCALEX                                                            41620
      D = D * SCALEX * SCALEW                                                   41630
      G = G * SCALEX * SCALEW                                                   41640
      U = C/B                                                                   41650
      H = F - E * U                                                             41660
      IF(H.EQ.ZERODO) H = SMALL                                                 41670
      ZC = D/B                                                                  41680
      Q =  (G-E*ZC)/H                                                           41690
      P = ZC - U*Q                                                              41700
      IF(Q.EQ.ZERODO) Q =-SMALL                                                 41710
      ZB = ONEDO/Q                                                              41720
      ZA = -P*ZB                                                                41730
      VREAL = PTFIVE * ZA                                                       41740
      RAD = ZA * ZA + 4.DO * ZB                                                 41750
      IF(RAD.GE.ZERODO) $FLAG = .TRUE.                                          41760
      IF(RAD.LT.ZERODO) RAD = -RAD                                              41770
      VIMAG = PTFIVE * DSQRT(RAD)                                               41780
```

```
      IF(DABS(VREAL).LT.(1.D-13)) VREAL = SMALL                          41790
      IF(DABS(VIMAG).LT.(1.D-13)) VIMAG = SMALL                          41800
      DELVR = DABS((DABS(VREAL)-DABS(OLDVR))/VREAL)                      41810
      OLDVR = VREAL                                                      41820
      DELVI = DABS((DABS(VIMAG)-DABS(OLDVI))/VIMAG)                      41830
      OLDVI = VIMAG                                                      41840
       VREAL = VREAL + DBLE(ATSHF)                                       41850
      IF(NJ.GT.ITR2) CALL VECOUT(16,&5)                                  41860
      IF(DELVR.GT.EPSVIB) GO TO 5                                        41870
      IF(DELVI.GT.EPSVIB) GO TO 5                                        41880
      IF($FLAG) GO TO 8                                                  41890
      CALL VECOUT(18,&6)                                                 41900
    5 CONTINUE                                                           41910
      CALL VECOUT(16,&7)                                                 41920
    8 TEMP = VIMAG                                                       41930
      VIMAG = VREAL - TEMP                                               41940
      VREAL = VREAL + TEMP                                               41950
       CALL VECOUT(17,&6)                                                41960
    6 RETURN                                                             41970
    7 RETURN 1                                                           41980
      END                                                                41990


      SUBROUTINE INITAA                                                  42000
C*****************************************************************      42010
C     THIS SUBROUTINE SETS THE OPERATING PARAMETERS                *     42020
C     OF THE PROGRAM FOR THE VIBRATION ANALYSES.                   *     42030
C*****************************************************************      42040
      IMPLICIT LOGICAL*1 ($)                                             42050
      REAL*8 ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,EPSAIT,  42060
     1       TWOPI,WV,XV,YV                                              42070
      COMMON /IBL1/ MNMAX                                                42080
      COMMON /IBL2/ N(10),MNINIT                                         42090
      COMMON /IBL4/ KMAX,KL                                              42100
      COMMON /IBL5/ IBCINL,IBCFNL                                        42110
      COMMON/IBL7/ MNMAXO,MAXD(10),MAXS(10),MAXSY(10),MODVIB(3,100),ITEM 42120
     1P(110)                                                             42130
      COMMON /IBL9/ MAXM                                                 42140
      COMMON /IBL10/ IFREC,NTRMAX                                        42150
      COMMON /IBL12/ KMAX1,KMAX2,NCONV                                   42160
      COMMON /IBL13/ ITRMAX,LSMAX                                        42170
      COMMON /BL6/ SOE,OSE,ALOAD                                         42180
      COMMON /BL16/ EPS                                                  42190
      COMMON/BL24/DG(4,4,4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12) 42200
      COMMON /BL100/ TEEC,$DYNMC                                         42210
      COMMON /BL102/ DELOAD                                              42220
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100) 42230
      COMMON /BLK1/ VMASS,NN2,$VIBES                                     42240
```

```
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGC,ISTOP                          42250
      COMMON /BLK4/ ZERODO,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL          42260
      COMMON /BLK8/ EPSVIB,EPSVEC,EPSAIT                                 42270
      COMMON /BLK9/ ITR1,ITR2                                            42280
      COMMON /BLK10/ ASHIFT,ASHFUM,ATSHF,BII,TOLUP,TOLDWN,TOLCLS         42290
      COMMON /BLK11/ DIAG(4)                                             42300
      COMMON /BLK12/NTRY                                                 42310
      COMMON/BLK13/DIF1,DIF2,DIF3,DIF4,TOLLOW,TOLHI,TOLRNG               42320
      COMMON /BLPLOT/ KLI(29),$PLOTS,$MODAL                              42330
      COMMON /BLDATA/ TITLE,NC,IMODE,NDIMEN,IPRINT,LCHMAX,IC             42340
      DIMENSION TITLE(18)                                                42350
C*****************************************************************     42360
C     THE FOLLOWING BLOCK OF STATEMENTS SETS THE CONSTANTS FOR      *    42370
C     THE DOUBLE PRECISION PORTION OF THE ITERATION CALCULATIONS.   *    42380
C*****************************************************************     42390
      ONEDO = 1.D0                                                       42400
      PTFIVE = .5D0                                                      42410
      SMALL = 1.0D-20                                                    42420
      TWOPI = 6.28318530717959D0                                         42430
      ZERODO = 0.D0                                                      42440
C*****************************************************************     42450
C     THE FOLLOWING BLOCK OF STATEMENTS SETS THE                    *    42460
C     ITERATION AND CONVERGENCE PARAMETERS.                         *    42470
C*****************************************************************     42480
      EPSAIT = 1.D-02                                                    42490
      EPSVIB = DBLE(EPS)                                                 42500
      IF(EPSVEC.EQ.ZERODO) EPSVEC = DBLE(EPS)                            42510
      IF($PLOTS.AND.(KMAX.GT.100))  KMAX=100                             42515
      KL = KMAX - 1                                                      42520
      KMAX1 = KMAX + 1                                                   42530
      KMAX2 = KMAX + 2                                                   42540
      IBEGIN = 1                                                         42550
      IF(IBCINL .LT. 0) IBEGIN = 2                                       42560
      IGO = ( IBEGIN * 4 ) - 3                                           42570
      IEND = KMAX2                                                       42580
      IF(IBCFNL .LT. 0) IEND = KMAX1                                     42590
      ISTOP = IEND * 4                                                   42600
      ITR1 = ITRMAX                                                      42610
      IF(ITR2.EQ.0) ITR2 = ITR1 - 2                                      42620
      NN2 = 2                                                            42630
      DIAG(4) = 0.0                                                      42640
      DO 1 I = 1,4                                                       42650
    1 WV(I,203) = DBLE(DIAG(I))                                          42660
C*****************************************************************     42670
C     THE FOLLOWING BLOCK OF STATEMENTS SETS THE OPERATING PARAMETERS   *42680
C     FOR THE FREQUENCY RANGE SEARCH  MODE OF OPERATION.            *    42690
C*****************************************************************     42700
      TOLCLS = .4                                                        42710
```

```
      TOLHI = 1.00500                                                   42720
      TOLLOW = 0.99500                                                  42730
      TOLRNG = 1.0 - (0.4 * EPS)                                        42740
      TOLSHF = 0.90                                                     42750
      TOLUP = 0.10                                                      42760
      NTRY = 1 + IFIX(1.0/TOLUP)                                        42770
C*****************************************************************    42780
C     THE FOLLOWING BLOCK OF STATEMENTS SETS THE OPERATING PARAMETERS  *  42790
C     FOR THE REST OF THE PROGRAM TO TREAT THE VIBRATION ANALYSIS AS A *  42800
C     SINGLE MODE LINEAR PROBLEM WITH THE OUTPUT TO BE IN MODAL FORMAT.*  42810
C*****************************************************************    42820
      ALOAD = 0.0                                                       42830
      DELOAD = 0.0                                                      42840
      $DYNMC = .TRUE.                                                   42850
      IC = 0                                                            42860
      IF($PLOTS) IFREQ = 1                                              42865
      IPRINT = 1                                                        42870
      ITRMAX = 1                                                        42880
      DO 2 I = 10,16                                                    42890
    2 KLI(I) = 0                                                        42900
      LCHMAX = 1                                                        42910
      LSMAX = 1                                                         42920
      MAXM = 1                                                          42930
      MNINIT = 1                                                        42940
      MNMAX = 1                                                         42950
      MNMAXO = 1                                                        42960
      $MODAL = .TRUE.                                                   42970
      NTHMAX = 0                                                        42980
C*****************************************************************    42990
C     THE FOLLOWING PARAMTERS ARE NOT PRESENTLY USED IN THE PROGRAM.   *  43000
C*****************************************************************    43010
      BII = 0.0                                                         43020
      TOLDWN = 0.0                                                      43030
      RETURN                                                            43040
      END                                                               43050

      SUBROUTINE VECOUT(IND,*)                                          43060
C*****************************************************************    43070
C     THIS SUBROUTINE CONTROLS THE PRINTED OUTPUT FOR THE FREE         *  43080
C     VIBRATION ANALYSIS PORTION OF THE PROGRAM. IN THE CASE OF A      *  43090
C     CONVERGED REAL EIGENVALUE(S) (E.G. IND=9) THE SOLUTION IS TESTED *  43100
C     FOR THE EXISTANCE OF AN IMAGINARY FREQUENCY (VREAL IS POSITIVE), *  43110
C     CALCULATES THE FREQUENCY IN CPS AND MAPS THE DOUBLE PRECISION    *  43120
C     EIGENVECTOR  WV(J,K) INTO THE SINGLE PRECISION VECTOR Z(J,K) FOR *  43130
C     USE IN SUBROUTINE OUTPUT.                                        *  43140
C     VARIOUS OTHER MESSAGES ARE PRINTED ACCORDING TO THE PARTICULAR   *  43150
C     SITUATION.                                                       *  43160
```

```
C*****************************************************************************
      IMPLICIT LOGICAL*1 ($)
      REAL*8 TWOPI,ZERODC,PTFIVE,ONEDC,SMALL,VREAL,VIMAG,EPSVIB,EPSVEC,
     1       EPSAIT,WV,XV,YV,WT,SCALEW,SCALEX,SCALE3
      COMMON/IBL7/ MNMAXO,MAXO(10),MAXS(10),MAXSY(10),MODVIB(3,100),ITEM
     1P(110)
      COMMON/BL24/DG(4,4.4),RANGE(100,4),TOP,BOTTOM,DELSHF,TOLSHF,RZ(12)
      COMMON /BL100/ TEEC,$DYNMC
      COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)
      COMMON /BLK2/ IVIBES,IVB
      COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP
      COMMON /BLK4/ ZERODC,PTFIVE,ONEDO,SMALL,VREAL,VIMAG,AREAL
      COMMON /BLK5/ $FLAG,$TAG,$DOVEC
      COMMON /BLK6/ NJ,NK,I4,IRGOLD,NUMBER,ITWO
      COMMON/BLK7/SCALEX,SCALEW,SCALE3
      COMMON /BLK8/ EPSVIB,EPSVEC,EPSAIT
      COMMON/BLK9/ITR1,ITR2
      COMMON /BLK11/ DIAG(4)
      COMMON /BLK12/NTRY
      COMMON /BLDATA/ TITLE,NC,IMODE,NDIMEN,IPRINT,LCHMAX,IC
      DIMENSION WT(812),Z(880),TITLE(18)
      EQUIVALENCE (WV(1,1),WT(1)),(WV(1,1),Z(1))
C*****************************************************************************
      ITEST = 0
      IF(IND.GE.7 .AND. IND.LE.9) GO TO 22
      IF(IND.EQ.17) GO TO 23
      GO TO 24
   22 IF(VREAL .GT. ZERODC) ITEST = 1
      IF(VREAL .LT. ZERODC) VREAL = -VREAL
      VREAL = DSQRT(VREAL) / (DBLE(TEEO) * TWOPI)
      GO TO 24
   23 IF(VREAL .GT. ZERODC) ITEST = 1
      IF(VIMAG .GT. ZERODC) ITEST = 2
      IF(VREAL .LT. ZERODC) VREAL = -VREAL
      IF(VIMAG .LT. ZERODO) VIMAG = -VIMAG
      VREAL = DSQRT(VREAL) / (DBLE(TEEO) * TWOPI)
      VIMAG = DSQRT(VIMAG) / (DBLE(TEEO) * TWOPI)
C*****************************************************************************
   24 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21),IND
C*****************************************************************************
C     SLOT 13 IS AVAILABLE FOR FUTURE USE.                                  *
C*****************************************************************************
    1 WRITE(6,201) MODVIB(2,IVB)
      WRITE(6,202) SHFTPT(1,IVB),SHFTPT(2,IVB),BOTTOM,TOP
      WRITE(6,203) DIAG
      IF($DOVEC) WRITE(6,207) EPSVEC
      IF(.NOT.$DOVEC) WRITE(6,208)
      RETURN 1
```

```
201 FORMAT('1',42X,'THIS IS A FREQUENCY RANGE SEARCH FOR HARMONIC',I3)
202 FORMAT('0',24X,'BOTTOM OF RANGE IS',E12.4,' HZ ',12X,'TOP OF RANGE
   3 IS',E12.4,' HZ '/35X,'BOTTOM =',E12.4,26X,'TOP =',E12.4)
203 FORMAT('0',41X,'DIAGONAL OF MASS MATRIX (MU-BAR) IS (',4F3.0,' )')
207 FORMAT('0',40X,'THE CONVERGENCE TEST ON THE VECTOR WILL BE PERFORM
   1ED'/50X,'CONVERGENCE CRITERION IS',E10.2)
208 FORMAT('0',38X,'THE CONVERGENCE TEST ON THE VECTOR WILL NOT BE PER
   2FORMED')
  2 WRITE(6,204) ITR1
    RETURN 1
204 FORMAT('0',44X,'CHECK PERMISSIBLE NUMBER OF ITERATIONS (',I2,')')
  3 WRITE(6,205) EPSVIB
    RETURN 1
205 FORMAT('0',51X,'EPS IS VERY SMALL (',E9.2,')')
  4 WRITE(6,206)
    RETURN 1
206 FORMAT('0',55X,'CHECK RANGE END POINTS')
  5 WRITE(6,219) MODVIB(2,IVB)
    WRITE(6,203) DIAG
    IF($DOVEC) WRITE(6,207) EPSVEC
    IF(.NOT.$DOVEC) WRITE(6,208)
    RETURN 1
219 FORMAT('1',41X,'THE LOWEST FREQUENCY IN HARMONIC',I4,' IS DETERMIN
   6ED')
  6 WRITE(6,220) MODVIB(1,IVB),MODVIB(2,IVB)
    WRITE(6,203) DIAG
    IF($DOVEC) WRITE(6,207) EPSVEC
    IF(.NOT.$DOVEC) WRITE(6,208)
    RETURN 1
220 FORMAT('1',41X,I2,' SHIFT POINT(S) FOR HARMONIC',I3,' IS(ARE) PROV
   7IDED')
  7 WRITE(6,209)NK
    WRITE(6,210) VREAL
    IF(ITEST .EQ. 1) WRITE(6,221)
    WRITE(6,224) SCALEW,SCALEX,SCALE3
    NJ = 0
    RETURN 1
209 FORMAT('0',36X,'NO CONVERGENCE TO A REAL EIGENVALUE AFTER ',I5,' I
   1TERATIONS')
210 FORMAT('0',38X,'RAYLEIGH QUOTIENT FREQUENCY =',E23.14,' HZ')
221 FORMAT('0',38X,'THE FREQUENCY IS IMAGINARY.')
224 FORMAT('0',34X,'THE LAST,NEXT TO LAST AND NEXT TO NEXT TO LAST EIG
   5ENVALUES ARE'//35X,3D21.12)
  8 WRITE(6,209) NK
    WRITE(6,210) VREAL
    IF(ITEST .EQ. 1) WRITE(6,221)
    WRITE(6,224) SCALEW,SCALEX,SCALE3
    RETURN 1
```

```
  9 WRITE(6,211) NK                                                      44120
    WRITE(6,210) VREAL                                                   44130
    IF(ITEST .EQ. 1) WRITE(6,221)                                        44140
    DO 91 I = IGO,ISTOP                                                  44150
 91 Z(I) = SNGL(WT(I))                                                   44160
    CALL OUTPUT(IMODE)                                                   44170
    RETURN 1                                                             44180
211 FORMAT('0',47X,'CONVERGED SOLUTION. ITERATIONS =',I6)                44190
 10 WRITE(6,301) BOTTOM                                                  44200
    RETURN 1                                                             44210
301 FORMAT('0',50X,'THE RANGE SIZE HAS BEEN REDUCED.'/51X,'BOTTOM OF R   44220
   1ANGE IS NOW',E14.6)                                                  44230
 11 WRITE(6,302)                                                         44240
    RETURN 1                                                             44250
302 FORMAT('0',55X,'RANGE HAS BEEN COVERED')                             44260
 12 WRITE(6,303) NTRY                                                    44270
    RETURN 1                                                             44280
303 FORMAT('0',38X,'UNABLE TO DETERMINE A MINIMUM  FREQUENCY AFTER ',I   44290
   13,' TRIES.')                                                         44300
 13 CONTINUE                                                             44310
    RETURN 1                                                             44320
 14 WRITE(6,305) NTRY                                                    44330
    RETURN 1                                                             44340
305 FORMAT('0',38X,'UNABLE TO DETERMINE A MAXIMUM  FREQUENCY AFTER ',I   44350
   13,' TRIES.')                                                         44360
 15 WRITE(6,304) TOP                                                     44370
    RETURN 1                                                             44380
304 FORMAT('0',50X,'THE RANGE SIZE HAS BEEN REDUCED.'/51X,'TOP OF RANG   44390
   2E IS NOW',E14.6)                                                     44400
 16 WRITE(6,212) NK,VREAL,VIMAG                                          44410
    WRITE(6,225) SCALEW,SCALEX,SCALE3                                    44420
    NJ = 0                                                               44430
    RETURN 1                                                             44440
212 FORMAT('0',47X,'NO CONVERGENCE AFTER',I6,' ITERATIONS.'/42X,'EIGEN   44450
   5VALUE(S) ARE',2E15.6)                                                44460
225 FORMAT('0',29X,'THE LAST, NEXT TO LAST AND NEXT TO NEXT TO LAST VE   44470
   4CTOR ELEMENT MAXIMUMS ARE'//35X,3D21.12)                             44480
 17 WRITE(6,213)                                                         44490
    WRITE(6,214) NK,VREAL,VIMAG                                          44500
    IF(ITEST .EQ. 1) WRITE(6,222)                                        44510
    IF(ITEST .EQ. 2) WRITE(6,223)                                        44520
    RETURN 1                                                             44530
213 FORMAT('0',45X,'CONVERGED SOLUTION - TWO REAL EIGENVALUES'/21X,'AN   44540
   1 INCREASE IN THE CONVERGENCE CRITERIA(EPS) AND/OR AN INCREASE IN T   44550
   2HE PERMISSIBLE NUMBER OF'/16X,'ITERATIONS(ITRMAX) IS RECOMMENDED A   44560
   3S A POSSIBLE SOLUTION TO THE PROBLEM OF A POORLY CONDITIONED MATRI   44570
   4X')                                                                  44580
214 FORMAT('0',29X,'ITERATIONS ARE',I6,10X,'FREQUENCIES, IN HZ, ARE',2   44590
```

```
     8E16.6)                                                                    44595
 222 FORMAT('0',63X,'THE FIRST FREQUENCY IS IMAGINARY.')                        44600
 223 FORMAT('0',63X,'BOTH FREQUENCIES ARE IMAGINARY.')                          44610
  18 WRITE(6,215) NK,VREAL,VIMAG                                                44620
     RETURN 1                                                                   44630
 215 FORMAT('0',46X,'CONVERGED SOLUTION - COMPLEX EIGENVALUE'/34X,'ITER         44640
    1ATIONS ARE',I6,10X,'EIGENVALUE IS',4X,2E15.6)                              44650
  19 WRITE(6,216) NUMBER                                                        44660
     RETURN 1                                                                   44670
 216 FORMAT('0',51X,I5,' FREQUENCIES HAVE BEEN FOUND')                          44680
  20 WRITE(6,217) BOTTOM,RANGE(1,1),RANGE(1,2)                                  44690
     RETURN 1                                                                   44700
 217 FORMAT('0',47X,'BOTTOM =',E12.4/48X,'LOWEST EIGENVALUE (RANGE(1,1)         44710
    2) =',E14.6/48X,'RANGE(1,2) =',E12.4)                                       44720
  21 WRITE(6,218) TOP,RANGE(1,4),RANGE(1,3)                                     44730
     RETURN 1                                                                   44740
 218 FORMAT('0',47X,'TOP =',E12.4/48X,'HIGHEST EIGENVALUE (RANGE(1,4))          44750
    3=',E14.6/48X,'RANGE(1,3) =',E12.4)                                         44760
     END                                                                        44770

     SUBROUTINE START(K)                                                        44780
C*****************************************************************************44790
C    SUBROUTINE START INITIALIZES THE EIGENVALUES AND EIGENVECTORS     *        44800
C    PRIOR TO THE START OF EACH ITERATION.                             *        44810
C*****************************************************************************44820
     REAL*8 TWOPI,ZERODD,PTFIVE,ONEDD,SMALL,VREAL,VIMAG,SCALEX,SCALEW,          44830
    1SCALE3,WV,WT,XV,XT,YV,YT,TEMP                                              44840
     COMMON/BL104/ WV(4,203),XV(4,203),YV(4,203),FILL(28),SHFTPT(3,100)         44850
     COMMON /BLK3/ TWOPI,IBEGIN,IEND,IGO,ISTOP                                  44860
     COMMON /BLK4/ ZERODD,PTFIVE,ONEDD,SMALL,VREAL,VIMAG,AREAL                  44870
     COMMON/BLK6/NJ,NK,I4,IROLD,NUMBER,ITWO                                     44880
     COMMON /BLK7/ SCALEX,SCALEW,SCALE3                                         44890
     DIMENSION WT(812),XT(812),YT(812)                                          44900
     EQUIVALENCE (WV(1,1),WT(1)),(XV(1,1),XT(1)),(YV(1,1),YT(1))                44910
C*****************************************************************************44920
     NJ = 0                                                                     44930
     NK = 0                                                                     44940
     ITWO = 1                                                                   44950
     SCALEW = ZERODD                                                            44960
     SCALEX = ZERODD                                                            44970
     SCALE3 = ZERODD                                                            44980
     VIMAG = ZERODD                                                             44990
     GO TO (1,3,3),K                                                            45000
   1 DO 2 I = IGO,ISTOP                                                         45010
   2 WT(I) = ONEDD                                                              45020
     GO TO 6                                                                    45030
   3 TEMP = ZERODD                                                              45040
```

```
      DO 4 I = IGO,ISTOP                                                    45050
    4 TEMP = DMAX1(DABS(YT(I)),TEMP)                                        45060
      IF(TEMP .LT. (1.D-40)) GO TO 1                                        45070
      TEMP = ONEDO / TEMP                                                   45080
      DO 5 I = IGO,ISTOP                                                    45090
    5 WT(I) = YT(I) * TEMP                                                  45100
    6 DO 7 I = IGO,ISTOP                                                    45110
    7 XT(I) = ONEDO                                                         45120
      RETURN                                                                45130
      END                                                                   45140

      SUBROUTINE MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISCALE)            45150
C*****************************************************************        45160
C         THIS SUBROUTINE SOLVES THE MATRIX EQUATION AX=B, WHERE A IS   *   45170
C         A SQUARE COEFFICIENT MATRIX AND B IS A MATRIX OF CONSTANT VEC-*   45180
C         TORS.                                                         *   45190
C         A(INVERSE) IS ALSO OBTAINED AND THE DETERMINANT OF A IS AVAIL-*   45200
C         ABLE.                                                         *   45210
C         THE FOLLOWING MUST BE DIMENSIONED IN THE CALLING PROGRAM:     *   45220
C         IPIVOT(N MAX), INDEX(N MAX,2), A(N MAX,N MAX), B(N MAX,N MAX) *   45230
C         WHERE:                                                        *   45240
C                    A = NAME OF 2-DIMENSIONAL ARRAY TO BE INVERTED     *   45250
C                    N = ORDER OF A -  1<=N<=NMAX                       *   45260
C                    B = NAME OF 2-DIMENSIONAL ARRAY TO BE MULTIPLIED   *   45270
C                          BY A(INVERSE)                                *   45280
C                    M = NUMBER OF COLUMN VECTORS IN B                  *   45290
C                          NOTE: M = 0 SIGNALS INVERSION ONLY)          *   45300
C                    IPIVOT = TEMPORARY STORAGE BLOCK                   *   45310
C                    INDEX = TEMPORARY STORAGE BLOCK                    *   45320
C                    NMAX = MAXIMUM ORDER OF A (AS DIMENSIONED IN THE   *   45330
C                          CALLING PROGRAM)                             *   45340
C                    DETERM= VALUE OF DETERMINANT AS GIVEN BELOW        *   45350
C                    ISCALE = USED IN FORMULA BELOW                     *   45360
C                                                                       *   45370
C                    DETERMINANT(A) = (10**18)**ISCALE*(DETERM)         *   45380
C                                                                       *   45390
C                    A(INVERSE) IS STORED IN A                          *   45400
C                    A(INVERSE)*B IS STORED IN B                        *   45410
C*****************************************************************        45420
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)                 45430
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)             45440
C*****************************************************************        45450
C     INITIALIZATION                                                        45460
C                                                                           45470
    5 ISCALE=0                                                              45480
    6 R1=10.0**18                                                           45490
    7 R2=1.0/R1                                                             45500
```

```
   10 DETERM=1.0                                                        45510
   15 DO 20 J=1,N                                                       45520
   20 IPIVOT(J)=0                                                       45530
   30 DO 550 I=1,N                                                      45540
C                                                                       45550
CC      SEARCH FOR PIVOT ELEMENT                                        45560
C                                                                       45570
   40 AMAX=0.0                                                          45580
   45 DO 105 J=1,N                                                      45590
   50 IF (IPIVOT(J)-1) 60, 105, 60                                      45600
   60 DO 100 K=1,N                                                      45610
   70 IF (IPIVOT(K)-1) 80, 100, 740                                     45620
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100                              45630
   85 IROW=J                                                            45640
   90 ICOLUM=K                                                          45650
   95 AMAX=A(J,K)                                                       45660
  100 CONTINUE                                                          45670
  105 CONTINUE                                                          45680
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1                                   45690
C                                                                       45700
CC      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL               45710
C                                                                       45720
  130 IF (IROW-ICOLUM) 140, 260, 140                                    45730
  140 DETERM=-DETERM                                                    45740
  150 DO 200 L=1,N                                                      45750
  160 SWAP=A(IROW,L)                                                    45760
  170 A(IROW,L)=A(ICOLUM,L)                                             45770
  200 A(ICOLUM,L)=SWAP                                                  45780
  205 IF(M) 260, 260, 210                                               45790
  210 DO 250 L=1, M                                                     45800
  220 SWAP=B(IROW,L)                                                    45810
  230 B(IROW,L)=B(ICOLUM,L)                                             45820
  250 B(ICOLUM,L)=SWAP                                                  45830
  260 INDEX(I,1)=IROW                                                   45840
  270 INDEX(I,2)=ICOLUM                                                 45850
  310 PIVOT=A(ICOLUM,ICOLUM)                                            45860
C                                                                       45870
CC      SCALE THE DETERMINANT                                           45880
C                                                                       45890
 1000 PIVOTI=PIVOT                                                      45900
 1005 IF(ABS(DETERM)-R1)1030,1010,1010                                  45910
 1010 DETERM=DETERM/R1                                                  45920
      ISCALE=ISCALE+1                                                   45930
      IF(ABS(DETERM)-R1)1060,1020,1020                                  45940
 1020 DETERM=DETERM/R1                                                  45950
      ISCALE=ISCALE+1                                                   45960
      GO TO 1060                                                        45970
 1030 IF(ABS(DETERM)-R2)1040,1040,1060                                  45980
```

```
 1040 DETERM=DETERM*R1                                                  45990
      ISCALE=ISCALE-1                                                   46000
      IF(ABS(DETERM)-R2)1050,1050,1060                                  46010
 1050 DETERM=DETERM*R1                                                  46020
      ISCALE=ISCALE-1                                                   46030
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070                                  46040
 1070 PIVOTI=PIVOTI/R1                                                  46050
      ISCALE=ISCALE+1                                                   46060
      IF(ABS(PIVOTI)-R1)320,1080,1080                                   46070
 1080 PIVOTI=PIVOTI/R1                                                  46080
      ISCALE=ISCALE+1                                                   46090
      GO TO 320                                                         46100
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320                                   46110
 2000 PIVOTI=PIVOTI*R1                                                  46120
      ISCALE=ISCALE-1                                                   46130
      IF(ABS(PIVOTI)-R2)2010,2010,320                                   46140
 2010 PIVOTI=PIVOTI*R1                                                  46150
      ISCALE=ISCALE-1                                                   46160
  320 DETERM=DETERM*PIVOTI                                              46170
C                                                                       46180
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT                                 46190
C                                                                       46200
  330 A(ICOLUM,ICOLUM)=1.0                                              46210
  340 DO 350 L=1,N                                                      46220
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT                                     46230
  355 IF(M) 380, 380, 360                                               46240
  360 DO 370 L=1,M                                                      46250
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT                                     46260
C                                                                       46270
C     REDUCE NON-PIVOT ROWS                                             46280
C                                                                       46290
  380 DO 550 L1=1,N                                                     46300
  390 IF(L1-ICOLUM) 400, 550, 400                                       46310
  400 T=A(L1,ICOLUM)                                                    46320
  420 A(L1,ICOLUM)=0.0                                                  46330
  430 DO 450 L=1,N                                                      46340
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T                                     46350
  455 IF(M) 550, 550, 460                                               46360
  460 DO 500 L=1,M                                                      46370
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T                                     46380
  550 CONTINUE                                                          46390
C                                                                       46400
C     INTERCHANGE COLUMNS                                               46410
C                                                                       46420
  600 DO 710 I=1,N                                                      46430
  610 L=N+1-I                                                           46440
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630                          46450
  630 JROW=INDEX(L,1)                                                   46460
```

```
  640 JCOLUM=INDEX(L,2)                                                  46470
  650 DO 705 K=1,N                                                       46480
  660 SWAP=A(K,JROW)                                                     46490
  670 A(K,JROW)=A(K,JCOLUM)                                              46500
  700 A(K,JCOLUM)=SWAP                                                   46510
  705 CONTINUE                                                           46520
  710 CONTINUE                                                           46530
  740 RETURN                                                             46540
      END                                                                46550

      SUBROUTINE FLYNVY                                                  46560
C*****************************************************************      46565
      WRITE (6,111)                                                      46570
      WRITE (6,1)                                                        46580
      WRITE (6,2)                                                        46590
      WRITE (6,3)                                                        46600
      WRITE (6,4)                                                        46610
      WRITE (6,5)                                                        46620
      WRITE (6,6)                                                        46630
      WRITE (6,7)                                                        46640
      WRITE (6,8)                                                        46650
      WRITE (6,9)                                                        46660
      WRITE (6,10)                                                       46670
      WRITE (6,11)                                                       46680
      WRITE (6,12)                                                       46690
      WRITE (6,13)                                                       46700
      WRITE (6,14)                                                       46710
      WRITE (6,15)                                                       46720
      WRITE (6,16)                                                       46730
      WRITE (6,17)                                                       46740
      WRITE (6,18)                                                       46750
      WRITE (6,19)                                                       46760
      WRITE (6,20)                                                       46770
      WRITE (6,21)                                                       46780
      WRITE (6,22)                                                       46790
      WRITE (6,23)                                                       46800
      WRITE (6,24)                                                       46810
      WRITE (6,25)                                                       46820
      WRITE (6,26)                                                       46830
      WRITE (6,27)                                                       46840
      WRITE (6,28)                                                       46850
      WRITE (6,29)                                                       46860
      WRITE (6,30)                                                       46870
      WRITE (6,31)                                                       46880
      WRITE (6,32)                                                       46890
      WRITE (6,33)                                                       46900
      WRITE (6,34)                                                       46910
```

```
      WRITE (6,35)
      WRITE (6,36)
      WRITE (6,37)
      WRITE (6,38)
      WRITE (6,39)
      WRITE (6,40)
  111 FORMAT ('1')
    1 FORMAT (' ',T11,'
     1          XXX')
    2 FORMAT (' ',T11,'
     1        XXXXX')
    3 FORMAT (' ',T11,'
     1     XXX    XXX')
    4 FORMAT (' ',T11,'
     1     XXX    XXX')
    5 FORMAT (' ',T11,'
     1        X@XXX')
    6 FORMAT (' ',T11,'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     1XXXXXXX@XXXXXXXXXXX        XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     2XX')
    7 FORMAT (' ',T11,'   XXXXXXXXOOXXXXXXXXOOXXXXXXXXXOOXXXXXXXXXOX
     1XXXXXXX@XXXXXXXXXXX        XOXXXXXXXXXOOXXXXXXXXXOOXXXXXXXOOXXXXXXXX'
     2)
    8 FORMAT (' ',T11,'          XXXXXXOOXXXXXXXOOXXXXXXXXOOXXXXXXOX
     1          @XXX              XOXXXXXXOOXXXXXXXXXOOXXXXXXXOOXXXXXX')
    9 FORMAT (' ',T11,'                 XXXXOOXXXXXXXOOXXXXXXXXXOOXXXOXX
     1      XXXXXXXXX             XXOXXXXOOXXXXXXXXOOXXXXXXXOOXXXX')
   10 FORMAT (' ',T11,'                             XXXXXXXXXXXXOOXXXXXXXOOXXXXXXXXXX
     1XXXXXXOOOOOOOXXXXXXXXXXXXXXXXXOOXXXXXXOOXXXXXXXXXXXX')
   11 FORMAT (' ',T11,'                                        XXXXXXXXXXXOOXXXXXXOOXXXXXXXOO
     1OOOOOOOOOOOOOOOOOOOOXXXXXXXOOXXXXXXOOXXXXXXXXXXXX')
   12 FORMAT (' ',T11,'                                                  XXXXXXXXXOOXXXXXXOOXXXXXOO
     1OOOOOOOOOOOOOOOOOOOOXXXXXOOXXXXXXOOXXXXXXXXX')
   13 FORMAT (' ',T11,'                                                       XXXXXXXOOXXXXXOOXXXXO
     1OOOOOOOOOOOOOOOOOOOXXXXOOXXXXOOXXXXXX')
   14 FORMAT (' ',T11,'                                                             XXXXOOXXXOOXXX
     1OX X X X X X X X XOOXXOOXXXOOXXXX')
   15 FORMAT (' ',T11,'                                                                 XXOXXXXOOX
     1OO X X X X X X X OOXOOXXXXOXX')
   16 FORMAT (' ',T11,'                                                                 OOOOOXXXXX
     1OO X X X X X X X OOXXXXXOOOOO')
   17 FORMAT (' ',T11,'                                                                  OOO
     1XOOX X X X X X XOOX      OOO')
   18 FORMAT (' ',T11,'                                                                   OOO
     1 XOO X X X X X OOX      OOO')
   19 FORMAT (' ',T11,'                                                                    OOO
     1  XOOX X X X XOOX      OOO')
   20 FORMAT (' ',T11,'                                                                     OOO
```

```
  1    XOO X X X OOX       OOO')
21 FORMAT (' ',T11,'                                                  OOOO
  1      XOO X OOX       OOOO')
22 FORMAT (' ',T11,'                                                   OOOO
  1       XOOOOOX       OOOO')
23 FORMAT (' ',T11,'                                                     OO
  1OOO  OOOXOXOOO  OOOOO')
24 FORMAT (' ',T11,'
  1 OOOOOOOOOOOOOOOO')
25 FORMAT (' ',T11,'
  1    OOOOOOOOOOOO')
26 FORMAT (' ',T11,'
  1      OOOOOOOO')
27 FORMAT (' ',T11,'
  1         OOO')
28 FORMAT (' ',//////////)
29 FORMAT (' ',T11,'       FFFFFFFFFFF  LLL        YYY          YYY
  1          NNN        NNN         A    VVV        VVV  YYY          YYY
  2')
30 FORMAT (' ',T11,'       FFFFFFFFFFF  LLL         YYY        YYY
  1          NNNN       NNN        AAA   VVV        VVV   YYY        YYY'
  2)
31 FORMAT (' ',T11,'       FFF          LLL          YYY      YYY
  1          NNNNN      NNN       AAAAA   VVV      VVV      YYY      YYY')
32 FORMAT (' ',T11,'       FFF          LLL           YYY   YYY
  1          NNNNNN     NNN       AAAAA   VVV      VVV       YYY   YYY')
33 FORMAT (' ',T11,'       FFF          LLL            YYY YYY
  1          NNNNNNN    NNN      AAA AAA   VVV   VVV         YYY YYY')
34 FORMAT (' ',T11,'       FFFFFFFF     LLL             YYYYY
  1          NNN NNNN   NNN      AAA AAA   VVV   VVV          YYYYY')
35 FORMAT (' ',T11,'       FFFFFFFF     LLL              YYY
  1          NNN  NNNN NNN     AAA   AAA   VVV VVV           YYY')
36 FORMAT (' ',T11,'       FFF          LLL              YYY
  1          NNN   NNNNNNN     AAAAAAAAA   VVV VVV           YYY')
37 FORMAT (' ',T11,'       FFF          LLL              YYY
  1          NNN    NNNNNN   AAAAAAAAAAA   VVVVV           YYY')
38 FORMAT (' ',T11,'       FFF          LLL              YYY
  1          NNN     NNNNN   AAA     AAA   VVVVV           YYY')
39 FORMAT (' ',T11,'       FFF          LLLLLLLLLLL      YYY
  1          NNN       NNNN  AAA       AAA   VVV           YYY')
40 FORMAT (' ',T11,'       FFF          LLLLLLLLLLL      YYY
  1          NNN        NNN  AAA       AAA    V            YYY')
   RETURN
   END
```

LIST OF REFERENCES

1. National Aeronautics and Space Administration Report NASA CR-1987, A Computer Program for the Geometrically Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution, Theory and Users Manual, by Robert E. Ball, April 1972.

2. National Aeronautics and Space Administration Report NASA CR-909, A Geometrically Nonlinear Analysis of Arbitrarily Loaded Shells of Revolution, by Robert E. Ball, January, 1968.

3. Ryan, B.A., A Digital Computer Study of the Buckling of Actual Imperfect Cylinders - A Modification of Program SATANS - Theory of User's Manual for SATANS-I and SATANS-II, Ae.E. Thesis, Department of Aeronautics, Naval Postgraduate School, Monterey, California, Dec. 1972.

4. Sanders, J. Lyell, Jr., "Nonlinear Theories for Thin Shells", Quarterly Journal of Applied Mathematics, vol. 21, no. 1, p. 21-36, 1963.

5. Budiansky, B. and Radkowski, P., "Numerical Analysis of Unsymmetrical Bending of Shells of Revolution", Journal of the American Institute of Aeronautics and Astronautics, vol. 1, no. 8, p. 1833-1842, August, 1963.

6. Air Force Flight Dynamics Laboratory Technical Report AFFDL-TR-68-144, Static, Free Vibration, and Stability Analysis of Thin, Elastic Shells of Revolution, part II, by Arturs Kalnins, March, 1969.

7. Wilkinson, J.H., The Algebraic Eigenvalue Problem, Clarendon Press, 1965.

8. Bodewig, E., Matrix Calculus, 2nd ed., Part IV, North-Holland, 1959.

9. Froberg, Carl-Erik, Introduction to Numerical Analysis, p. 93-103, 2nd ed., Addison-Wesley, 1969.

10. Crandall, S.H., Ph.D., Engineering Analysis, Chaps. 4,5, McGraw-Hill, 1956.

11. Fox, L., An Introduction to Numerical Linear Algebra, Chap. 8, Oxford University Press, 1965

12. Meirovitch, L., Analytical Methods in Vibrations, Chap. 4, MacMillan Company, 1967.

13. Prentis, J.M. and Leckie, F.A., Mechanical Vibrations: An Introduction to Matrix Methods, Chap. 3, Longmans, Green and Company, Ltd., 1963.

14. Anderson, M.S., Fulton, R.E., Heard, W.L., Jr., and Walz, J.E., Stress, Buckling and Vibration Analysis of Shells of Revolution, paper presented at the Conference on Computer Oriented Analysis of Shell Structures, Palo Alto, California, 10 August, 1970.

15. Cohen, G.A., Computer Analysis of Axisymmetric Free Vibrations of Ring-Stiffened Orthotropic Shells of Revolution, Journal of The American Institute of Aeronautics and Astronautics, vol. 3, no. 12, p. 2305-2312, December, 1965.

16. Westlake, J.R., A Handbook of Numerical Matrix Inversion and Solution of Linear Equations, John Wiley and Sons, 1968.

17. Gregory, R.T. and Karney, D.L., A Collection of Matrices for Testing Computational Algorithms, Wiley-Interscience, 1969.

18. Moler, C.B. and Stewart, G.W., An Algorithm for the Generalized Matrix Eigenvalue Problem $Ax = \lambda Bx$, Stanford University Computer Science Department (STAN-CS-232-71), August, 1971.

19. INTERNATIONAL MATHEMATICAL AND STATISTICAL LIBRARIES, LIBRARY 1, IMSL, 2nd ed., July, 1972.

20. Kamel, H.A. and Lambert, R.L., Solution of Structural Eigenvalue Problems Using Sparsely Populated Matrices, Paper presented at the Conference on Computer Oriented Analysis of Shell Structures, 10 August, 1970.

21. Fulton, R.E., and Blum, R.E., A Modification of Potters' Method for Solving Eigenvalue Problems Involving Tridiagonal Matrices, Astronautics, vol. 4, no. 12, p. 2231-2232, December, 1966.

22. National Aeronautics and Space Administration Report NASA TN D-3831, Vibration and Buckling of Prestressed Shells of Revolution, by Paul A. Cooper, March, 1967.

23. National Aeronautics and Space Administration Report NASA SP-221, The NASTRAN Theoretical Manual, Sections 9 and 10, edited by Richard H. MacNeal, September 1970.

24. National Aeronautics and Space Administration Report NASA TN D-3844, Matrix Analysis of Longitudinal and Torsional Vibrations in Nonuniform Multibranch Beams, by Robert T. Wingate, February, 1967.

25. National Aeronautics and Space Administration Contractor Report NASA-CR-1316, A Theoretical Analysis of the Free Vibration of Discretely Stiffened Cylindrical Shells With Arbitrary End Conditions, by D. M. Egle and K.E. Soder, Jr. June 1969.

26. United States Naval Postgraduate School Report NPS-57BP7161A, Vibration Analysis of Cylindrical Shells by Several Finite Difference Schemes, by R.E. Ball, June 1971.

27. National Aeronautics and Space Administration Technical Note NASA-TN-D- 4972, A Method for Computation of Vibration Modes and Frequencies of Orthotropic Thin Shells of Revolution Having General Meridional Curvature, by H. M. Adelman, D.S. Catherines and W.C. Walton, Jr, January 1969.

28. National Aeronautics and Space Administration Technical Note NASA-TN-D-2767, Conical Shell Vibrations, by D. H. Platus, April 1965.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Professor Robert E. Ball Department of Aeronautics Naval Postgraduate School Monterey, California 93940	2
4.	Chairman, Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
5.	LT Ralph R. Nebiker 115 Malloway Lane Monterey, California 93940	2
6.	Professor Lars Åke Samuelson The Aeronautical Research Institute of Sweden P. O. Box 11021 S - 161 11 Bromma 11 Sweden	1
7.	Mr. Richard Citerly Anamet Laboratories P. O. Box 831 San Carlos, California 94070	1
8.	Dr. Robert Fulton Structural Mechanics Branch NASA - Langley Research Center Hampton, Virginia 23365	1
9.	LCDR Bruce A. Ryan Naval Air Systems Command Headquarters (Air-09PB) Washington, DC 20360	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3. REPORT TITLE

Natural Modes and Frequencies of Free Vibration of Shells of Revolution – Analysis of a Shell of Linearly Varying Thickness – SATANS-III: Theory and User's Manual

4. DESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; March 1973

5. AUTHOR(S) (First name, middle initial, last name)

Ralph Robert Nebiker

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1973	303	28

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A digital computer program known as SATANS-III is developed as a modification to the program SATANS-I, incorporating the capability to perform free vibration analyses for the natural frequencies and modes of thin, elastic shells of revolution. The geometric and material properties of the shell may vary along the meridian. SATANS-III retains the SATANS-1 capability of performing geometrically nonlinear static and dynamic analyses of arbitrarily loaded shells. The free vibration analysis employs the inverse iteration method with spectral shifts, including the capability to search a specific range of frequencies. Several example problems are solved to illustrate the program's validity and accuracy. A user's manual for SATANS-III is presented. In addition, the problem of handling a large, strongly banded, unsymmetric and occasionally singular stiffness matrix in conjunction with a singular mass matrix is treated. A discussion of several solution procedures with respect to this problem is given.

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31406

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
SHELLS						
SHELLS OF REVOLUTION						
FREQUENCIES						
NATURAL FREQUENCIES						
NATURAL FREQUENCIES OF SHELLS						
VIBRATION						
FREE VIBRATION						
FREE VIBRATION OF SHELLS						
MODES						
MODES OF VIBRATION						
NATURAL MODES OF VIBRATION						
NATURAL MODES OF VIBRATION OF SHELLS						
STRUCTURAL VIBRATIONS						
FREE VIBRATIONS OF STRUCTURES						
EIGENVALUES						
REAL EIGENVALUES						
UNSYMMETRIC MATRICES						
EIGENVALUES OF UNSYMMETRIC MATRICES						
BUCKLING OF SHELLS						
STRUCTURAL ANALYSIS						
SATANS						
SATANS-III						

Thesis
N3524
c.2

143418

Nebiker
Natural modes and frequencies of free vibration of shells of revolution - analysis of a shell of linearly varying thickness - Satans-III: Theory and user's manual.

Thesis
N3524
c.2

143418

Nebiker
Natural modes and frequencies of free vibration of shells of revolution - analysis of a shell of linearly varying thickness - Satans-III: Theory and user's manual.

thesN3524
Natural modes and frequencies of free vi
3 2768 002 01775 8
DUDLEY KNOX LIBRARY